Neural Radiance Field in Autonomous Driving: A Survey

Lei He[1], Leheng Li[2], Wenchao Sun[1], Zeyu Han[1], Yichen Liu[3], Sifa Zheng[1], Jianqiang Wang[1], Keqiang Li[1*]

***Abstract*—Neural Radiance Field (NeRF) has garnered significant attention from both academia and industry due to its intrinsic advantages, particularly its implicit representation and novel view synthesis capabilities. With the rapid advancements in deep learning, a multitude of methods have emerged to explore the potential applications of NeRF in the domain of Autonomous Driving (AD). However, a conspicuous void is apparent within the current literature. To bridge this gap, this paper conducts a comprehensive survey of NeRF's applications in the context of AD. Our survey is structured to categorize NeRF's applications in Autonomous Driving (AD), specifically encompassing perception, 3D reconstruction, simultaneous localization and mapping (SLAM), and simulation. We delve into in-depth analysis and summarize the findings for each application category, and conclude by providing insights and discussions on future directions in this field. We hope this paper serves as a comprehensive reference for researchers in this domain. To the best of our knowledge, this is the first survey specifically focused on the applications of NeRF in the Autonomous Driving domain.**

***Index Terms*—Neural Radiance Field, Autonomous driving, Perception, 3D Reconstruction, SLAM, Simulation**

I. Introduction

NERF, as an advanced novel view synthesis technology, harnesses the capabilities of volume rendering and implicit neural scene representation to unveil the complexity of 3D scene geometry. It made its debut at ECCV 2020 [1], rapidly achieving a leading level of visual quality and serving as a wellspring of inspiration for numerous subsequent research endeavors. In recent years, the domain of autonomous driving has made significant strides, with widespread deployment in highway scenarios, although deployment in urban environments is still undergoing rigorous testing. This technological evolution has shifted from its initial reliance on high-precision maps to provide static scene understanding, now emphasizing real-time perception of local environments through bird's-eye view vision. Simultaneously, it has progressed functionally from Level 2 (L2) and is striving towards Level 4 (L4) autonomy. Autonomous driving systems demand a deep understanding of the surrounding environment, encompassing both static scenes and the dynamic interactions among traffic participants, which is a critical prerequisite for effective planning and control. Through self-supervised learning, NeRF has demonstrated its ability to effectively comprehend local scenes, making it an enticing candidate for enhancing autonomous driving capabilities. Over the past two years, NeRF models have found applications in various aspects of autonomous driving, including perception, 3D reconstruction, simultaneously localization and mapping(SLAM), and simulation, as shown in Fig. 1.

Neural Radiance Fields (NeRF) has emerged as a promising contender in the field of perception, encompassing a range of critical tasks such as object detection, semantic segmentation, and occupancy prediction. The surge in popularity is primarily attributed to its exceptional ability to acquire precise and consistent geometric information. Research in this field can be classified into two main paradigms, differentiated by the utilization of NeRF: "NeRF for data" and "NeRF for model". The former involves the initial training of NeRF, followed by its use to augment the training data of perception tasks. In contrast, the latter adopts a collaborative training strategy for NeRF and perception networks, enabling the perception networks to learn the geometric information captured by NeRF.

In the realm of 3D reconstruction applications, NeRF can be categorized into three primary methods based on the level of scene understanding: dynamic scene reconstruction, surface reconstruction, and inverse rendering. In the first category, dynamic scene reconstruction focus on reconstructing the dynamic scenes with movable agents, mostly with the sequential 3D bounding box annotation and camera parameters. In the second category, surface reconstruction aims to reconstruct explicit 3D surfaces of the scenes, such as mesh. In the third category, inverse rendering aims to disentangle shape, albedo, and visibility from images of driving scenes, to enable applications such as relighting.

As for SLAM applications, the utilization of NeRF can be classified into three primary methods, each geared towards mapping, localization, or a combination of both. As for localization, NeRF is employed to perform real-time image rendering at the current timestamp and estimate the precise pose of the SLAM system by minimizing the reprojection error. While NeRF for mapping primarily focuses on enhancing the mapping capabilities of the SLAM system, which achieves this by incorporating depth maps generated using NeRF, resulting in improved map accuracy. Besides, NeRF is used in some other research to simultaneously enhance the quality of the 3D map and improve the SLAM system's accuracy in pose estimation. These categorizations demonstrate how NeRF can be strategically integrated into a SLAM system to meet specific needs, whether they involve mapping, localization, or a combination of both functionalities. It is worth mentioning that some of the existing NeRF-based SLAM approaches are

[1]School of Vehicle and Mobility, Tsinghua University, Beijing, China
[2] AI Thrust, Information Hub, The Hong Kong University of Science and Technology - Guangzhou Campus, Guangzhou, China
[3]School of Engineering Mathematics and Technology, University of Bristol, Bristol, UK
*Correspondence: likq@tsinghua.edu.cn

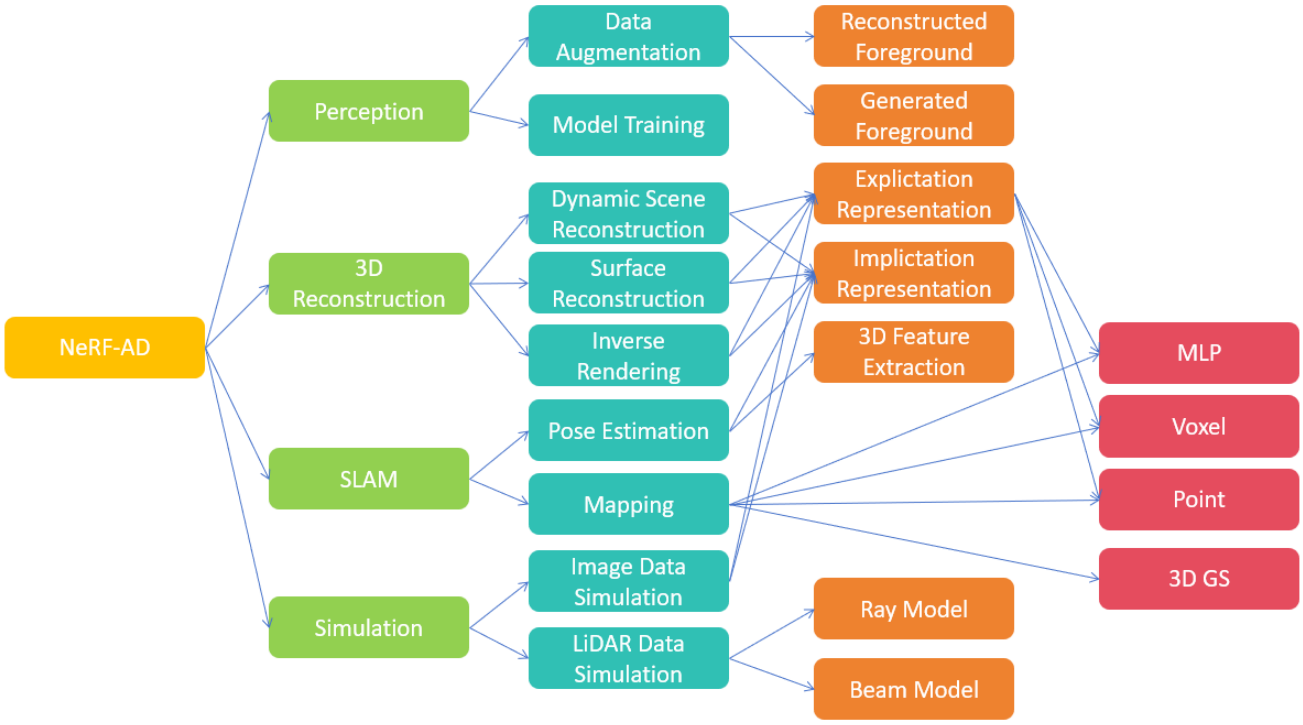

Fig. 1: A taxonomy of Neural Radiance Field in Autonomous Driving.

designed for indoor scenarios, but as the technique is similar to large-scale outdoor environment for autonomous driving, indoor approaches are also reviewed in this paper.

In NeRF simulations, there are two types. The first type divides driving scenes into static and dynamic components, using neural radiance fields for both. It then edits the motion of vehicles or pedestrians to generate new scenes and simulate image data. This type is further split into implicit and explicit approaches, depending on scene representation. The second type centers on simulating LiDAR data from new viewpoints, integrating LiDAR sensing process models with neural radiance fields to depict the scene's geometry. This type is divided into ray and beam models, based on the modeling differences of the LiDAR sensing process.

As remarkable advancements in both academic and industrial domains have unfolded in this field, we present a comprehensive review of recent developments to catalyze further research. The primary contributions of this work can be summarized as follows:

- To the best of our knowledge, this is the first comprehensive survey reviewing NeRF's applications in addressing fundamental techniques within the realm of autonomous driving.
- We provide the latest NeRF-AD methodologies, systematically categorizing them based on their core principles and downstream applications.
- We present a comprehensive discussion of NeRF-AD, offering insights into critical research gaps and suggestions for future research directions.

The structure of this paper is outlined as follows. Section II provides an introduction to the basic principles and background of NeRF. Section III delves into the analysis of NeRF applications in perception. Section IV conducts an in-depth comparative analysis of NeRF applications in 3D reconstruction. Section V offers a detailed analysis of NeRF applications in SLAM. Section VI provides an in-depth analysis of NeRF applications in simulation. Section VII discusses and predicts future research directions. Finally, Section VIII summarizes the paper.

II. Neural Radiance Field

Neural Radiance Fields, first introduced by Mildenhall et al. [1] in 2020, achieved highly realistic view synthesis of complex scenes using only 2D posed images for supervision. NeRF conceptualizes a continuous scene as a 5D vector-valued function. This 5D scene representation, facilitated through an MLP network, is denoted as:

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

where $\mathbf{x} = (x, y, z)$ denotes the coordinates of points within the scene, (θ, ϕ) refer to the azimuthal and polar viewing angles respectively, $c = (r, g, b)$ represents the color, and σ signifies the volume density. In practical implementations, (θ, ϕ) are represented as $\mathbf{d} = (d_x, d_y, d_z)$, a 3D Cartesian unit vector. The network architecture is structured in two stages, where the first stage inputs $\mathbf{x}$ and outputs σ along with a feature vector. In the second stage, this feature vector is concatenated with the viewing direction $\mathbf{d}$ to produce the color $\mathbf{c}$ at that viewpoint. This design enables the network to learn a view-independent σ that relies solely on the in-scene coordinates and a color $\mathbf{c}$ that depends on both the viewing direction and in-scene coordinates.

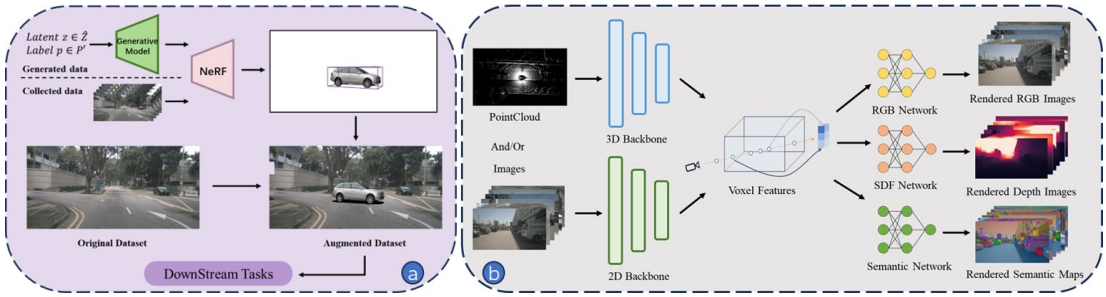

Fig. 2: Overview of NeRF's application in autonomous driving perception: (a) NeRF can be used for data augmentation by reconstructing scenes from either generated data or collected real data. (b) NeRF's implicit representation and neural rendering can be integrated into model training to enhance performance.

Given the volume density and color for each point in a scene, NeRF employs volume rendering, as described in [2], to compute the color $C(\mathbf{r})$ of any camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where $\mathbf{o}$ is the camera position and $\mathbf{d}$ is the viewing direction, using the following equation:

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

where $\sigma(\mathbf{r}(t))$ and $\mathbf{c}(\mathbf{r}(t), \mathbf{d})$ correspond to the volume density and color at point $\mathbf{r}(t)$ on the path of the camera ray oriented in direction $\mathbf{d}$, and dt indicates the incremental distance that the ray traverses at each step of the integration.

$T(t)$ represents the accumulated transmittance, indicating the probability that the ray travels from t_1 to t without being obstructed. This is defined as follows:

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

Novel views are synthesized by projecting camera rays $C(\mathbf{r})$ through each pixel of the target image, with the integral evaluated numerically. The original implementation and most subsequent works have employed a stochastic stratified sampling strategy, segmenting the ray into N uniform segments and selecting a random sample within each. Consequently, the rendering equation is approximated as:

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

where δ_i denotes the distance between the ith and $i+1$th sample points. (σ_i, c_i) are the density and color at the ith sample point along the given ray, as computed by the MLP. $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$ represents the opacity at the ith sample point.

To capture finer details in models, the NeRF framework often incorporates positional encoding. In the original paper, this technique, denoted as γ, is applied to each component of the scene coordinate $\mathbf{x}$ (which is normalized to the range $[-1, 1]$) and the viewing direction $\mathbf{d}$. This encoding enhances the model's ability to represent high-frequency functions.

For each pixel, the MLP parameters are optimized using a square error photometric loss. Across the entire image, this loss is calculated as follows:

$$L = \sum_{r \in R} ||\hat{C}(\mathbf{r}) - C_{gt}(\mathbf{r})||_2^2 \tag{5}$$

where $C_{gt}(\mathbf{r})$ represents the actual color of the pixel in the training image corresponding to ray $\mathbf{r}$, and R denotes the collection of rays linked to the image being synthesized.

To capture finer details in models, the NeRF framework often incorporates positional encoding. In the original paper, this technique, denoted as γ, is applied to each component of the scene coordinate $\mathbf{x}$ (which is normalized to the range $[-1, 1]$) and the viewing direction $\mathbf{d}$. This encoding enhances the model's ability to represent high-frequency functions. This encoding is calculated as follows:

$$\gamma(v) = (sin(2^0 \pi v), cos(2^0 \pi v), sin(2^1 \pi v), cos(2^1 \pi v), \\ ..., sin(2^{N-1} \pi v), cos(2^{N-1} \pi v)), \tag{6}$$

In the original paper, the positional encoding levels were set at $N = 10$ for $\mathbf{x}$ and $N = 4$ for $\mathbf{d}$, optimizing the model's sensitivity to spatial and directional variations.

III. Perception

NeRF demonstrate significant potential in autonomous driving perception tasks, which are categorized into two branches: data augmentation and model training. Data augmentation entails utilizing NeRF's innovative view synthesis capabilities to conduct photorealistic data augmentation for training datasets, while model training involves integrating neural rendering into the training process to capture geometric details and enhance performance. This paper delineates the pipelines of these two branches, as illustrated in Fig. 2.

A. Data Augmentation

Driving scenes are widely recognized for their remarkable diversity and complexity, making it infeasible to capture all scenarios due to the long-tail problem and high costs. Data augmentation stands as an effective technique to enrich training datasets and enhance model performance. Various studies[3–6] utilize graphic engines to synthesize training data, thereby introducing a sim-to-real domain gap. NeRF, however, exhibits a smaller domain gap as it is trained to approximate realistic images.

Drive-3DAug[7] pioneered research in 3D data augmentation for camera-based 3D perception and demonstrated that NeRF is an effective solution for this purpose. Unlike traditional 2D image augmentation techniques, which are limited

to operations on the image plane, such as rotation and copy-and-paste, 3D augmentation has the potential to significantly improve model performance, which has been witnessed in LiDAR-based 3D perception tasks. As shown in Fig. 3, Drive-3DAug comprises two stages: the initial training stage decomposes scenes into background and foreground and constructs 3D models using NeRF, which then serve as reusable digital assets. The subsequent stage involves combining the background with manipulated foreground to create new driving scenes and utilizes volume rendering to produce augmented images. Through 3D data augmentation, object detection models trained with NeRF-based augmentation exhibit superior performance compared to those trained with only 2D data augmentation.

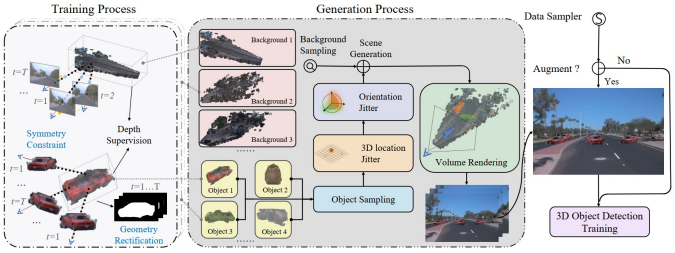

Fig. 3: The pipeline of Drive-3DAug[7].

Neural Radiance Fields (NeRF) are leveraged to reconstruct scenes using not only collected sensor data but also label-aware data synthesized by generative models, with the specific aim of reducing annotation costs. Lift3D[8] explores the combination of Generative Adversarial Networks (GAN) and Neural Radiance Fields (NeRF) with the aim of generating data for 3D perception tasks. Initially, pretrained StyleGAN2 is utilized to densely sample images with pose labels. It is assumed that the first 8 layers of the latent code control poses, while the remaining layers influence shape and appearance. A 3D car model from ShapeNet is used to obtain rendered car images from different viewpoints and their corresponding pose labels. Subsequently, an optimization-based GAN inversion method is employed to find the corresponding template latents of the first 8 layers, associating these template latent layers with meaningful pose information. The latent-pose pairs are then incorporated into a 3D shared conditional NeRF, following a 2D-to-3D pipeline. This process eliminates the need for a 2D upsampler, as required by previous methods, and enables the synthesis of images in any resolution. Finally, the trained NeRF can be used to render augmented images for downstream task training.

Based on Lift3D[8], Adv3D[9] present an innovative exploration of modeling adversarial examples within the context of NeRF, integrating primitive-aware sampling and semantic-guided regularization for 3D patch attacks with camouflage adversarial texture. Their approach involves training an adversarial NeRF to minimize the confidence of 3D detectors for surrounding objects in the training set, resulting in strong generalization capabilities across various poses, scenes, and 3D detectors. Additionally, the paper introduces a defense mechanism against these attacks, employing adversarial training through data augmentation. The intersection of adversarial examples and 3D modeling showcased in this work indicates potential implications for the security and robustness of 3D perception systems, offering valuable insights for applications including autonomous vehicles, robotics, and augmented reality.

B. Model Training

Several studies have investigated the use of NeRF for data augmentation, but there is a growing body of research that integrates NeRF representation into models to enhance performance. By harnessing implicit scene representation and neural rendering, NeRF effectively bridges the gap between 3D scenes and 2D images, makinging it suitable for a variety of 3D perception tasks.

NeRF has exhibited remarkable performance in scene reconstruction and consequently has found natural applications in perception tasks related to scene completion. BTS[10] were among the first to apply volume rendering in single-view reconstruction. Their approach involves inferring an implicit density field as a meaningful geometric scene representation instead of relying solely on depth prediction, which can only reason about visible areas in the image. They utilize an encoder-decoder network to predict a pixel-aligned feature map from the input image. To compute the density value at a given 3D point, features are bilinearly sampled from the feature maps after the 3D point is projected onto the image. Subsequently, along with the features, the depth value of this point and positional encoding are input into a multi-layer perceptron (MLP) to predict the density. The depth can be generated as a by-product of the density field, and for novel view synthesis, colors are sampled from other views rather than being predicted by the MLP, thus drastically reducing the complexity of the distribution along a ray, since density distributions tend to be simple. Multiple views, apart from the input view, are utilized in the training process. These views are divided into two sets, namely N_{loss} and N_{render}. Colors are sampled from N_{render} and then used to reconstruct N_{loss}, where the photometric consistency between the reconstructed view and N_{loss} serves as the training signal for the density field. This training strategy facilitates the ability to reason about occluded areas in the input view, provided they are visible to other views.

The reasoning of occluded areas is highly relevant to semantic scene completion, as investigated in the work S4C[11]. As shown in Fig. 4, the processing pipeline is based on BTS[10] but incorporates a semantic field in parallel with the density field, enabling the rendering of a semantic map. The disparity between the semantic map and the pseudo-ground truth labels obtained from an off-the-shelf segmentation network provides an additional training signal. Since supervision provided from a single viewpoint only offers training signals for observed areas, it is crucial to strategically select training views. Therefore, sideways-facing views with a random offset from the input view are chosen for training, thereby enhancing diversity and improving the quality of predictions, particularly for further away regions.

Being capable of capturing accurate geometry, NeRF can also be applied to occupancy prediction tasks.

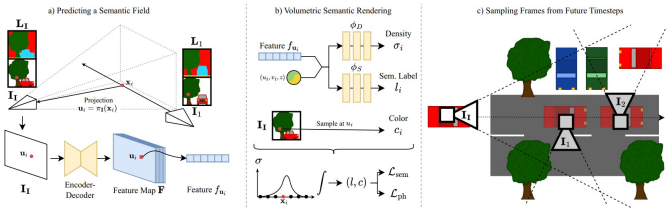

Fig. 4: The pipeline of S4C[11].

SimpleOccupancy[12] made an attempt at 3D occupancy estimation, focusing solely on geometry estimation and setting it apart from other similar works. They utilize a shared backbone to extract image features and then apply bilinear interpolation in a parameter-free manner to lift these features to 3D volume space. A 3D convolution network and position embedding are employed for 3D volume feature aggregation. Subsequently, the occupancy probability value is obtained by applying a Sigmoid function. The training process can be supervised using two manners: one involves directly computing a classification loss based on the occupancy probability, while the other utilizes volume rendering to obtain a depth map and supervises it against depth labels. The results indicate that depth loss outperforms classification loss across various metrics, demonstrating the effectiveness of volume rendering.

UniOcc[13] utilizes volume rendering to integrate 2D and 3D representation supervision. Similar to previous research, the approach involves the use of a 2D image encoder, 2D-3D view transformer, and a 3D encoder to generate 3D voxel features, as depicted in Fig. 5. However, unlike existing methods, UniOcc converts occupancy into NeRF-style representation instead of directly employing an occupancy head for occupancy estimation. It achieves this by using two separate MLPs to predict the density and semantic logits of the voxels. Subsequently, geometric and semantic rendering techniques are applied based on the density and semantic logits to generate 2D depth and semantic logits, which can be supervised by 2D labels. Given the sparsity of viewpoints, temporal frames are introduced as supplementary perspectives after filtering moving objects by semantic categories. As a result of the architectural design and various optimization techniques, UniOcc achieved a 3rd place ranking in the CVPR 2023 3D Occupancy Prediction Challenge.

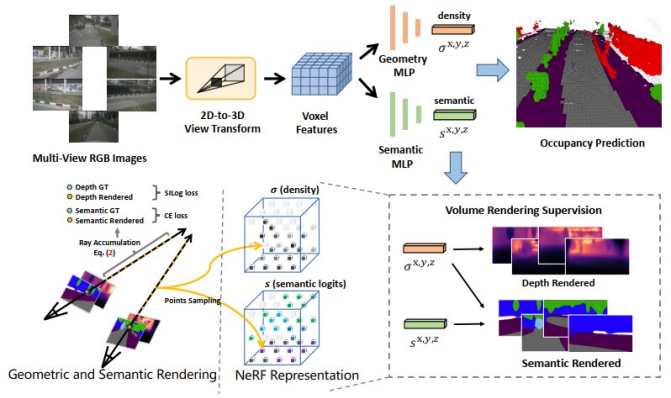

Fig. 5: The pipeline of UniOcc[13].

RenderOcc[14] demonstrated that 3D occupancy labels are not only expensive, but may also impede model performance due to the inherent ambiguity of occupancy annotation. This limitation constrains the usability and scalability of 3D occupancy models. As a result, they made the pioneering attempt to train 3D occupancy networks solely using 2D labels and achieved competitive performance compared to those supervised by 3D labels. The model architecture shares similarities with UniOcc[13] but takes a step forward by incorporating temporal frames. Auxiliary rays from adjacent frames are introduced to reinforce multi-view consistency, albeit at the expense of high memory and computational costs, necessitating ray sampling during training. Random sampling may lead to the discarding of many valuable rays, and rays from adjacent frames may cause misalignment due to the movement of dynamic objects. To address these issues simultaneously, a weighted ray sampling method is proposed. This method assigns lower probability weights in random sampling to rays associated with low-information-density backgrounds or dynamic objects, thereby reducing the likelihood of their being sampled. Consequently, this increases information density and mitigates temporal misalignment.

The NeRF model is also applied to the 3D detection task in MonoNeRD[15]. Utilizing scene geometry to improve the detector's performance is a common approach, and depth estimation has been widely adopted in previous work. However, this often results in sparsity in the 3D representations and significant information loss. In MonoNeRD, a NeRF-like representation is employed for dense 3D geometry. Initially, a camera frustum with multiple depth planes is constructed to extract image features in a Query-Key-Value manner. Subsequently, two convolution blocks transform the frustum features into SDF and RGB features, where the SDF features can be further converted to density features. These density and RGB features are utilized for volume rendering to supervise the model using depth loss and RGB loss. As the irregular frustum features cannot be directly used by downstream detection modules, 3D voxel features are constructed by trilinear sampling from the frustum features and then fed to the detection head. It is also noted that other views can be utilized for rendering as long as their frustum overlaps with the original one.

NeRF is particularly well-suited for static perception tasks, such as map construction, due to its inherent property of multi-view consistency. In contrast to current on-board map construction approaches, MV-Map[16] focuses on off-board HD-Map generation. The methodology involves using a pre-trained bird's-eye view (BEV) segmentation model to generate BEV features and semantic maps for each frame in a frame-centric manner. These are then aggregated globally in a region-centric manner. The BEV features serve as input to an uncertainty network, which generates confidence maps. The semantic content of a grid is determined by a weighted average of all semantic maps that overlap with it. A voxelized NeRF is employed to cover the entire scene and capture consistent multi-view geometry. Additionally, the study suggests that the predicted semantics at a position are more reliable when they reside to object surfaces. So for each voxel, the 3D position of its projected pixel location can be reconstructed by the trained

NeRF, and the residual between the voxel center and the 3D position is used as an augmented input for the uncertainty network, representing the proximity of the voxel to object surfaces.

Beyond specific image-based tasks, volume rendering has the potential to bridge the gap between point clouds and images, facilitating representation learning for pre-training. In their work, PRED[17] focuses on the pre-training of LiDAR point clouds. As depicted in Fig. 6, the authors first apply point-wise masking to the input point cloud, preserving the semantics of objects even in sparser regions. The remaining point cloud is then transformed into a Bird's Eye View (BEV) feature map by an encoder, which is subsequently mapped to Signed Distance Function (SDF) and semantic information by a decoder. Due to the absence of color information in the point cloud, only semantic and depth supervision are used after volume rendering. By leveraging semantic rendering, the comprehensive information and rich semantics of images enhance the performance of point cloud pre-training across various tasks.

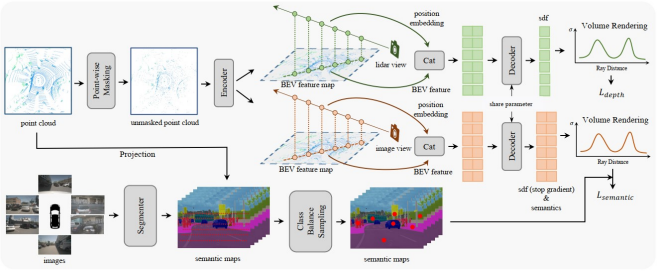

Fig. 6: The overall pipeline of PRED[17].

UniPAD[18], going a step further, propose a flexible pre-training method that enables seamless integration into both 2D and 3D frameworks. This method comprises two components: a modality-specific encoder and a volumetric rendering decoder. For point cloud data, a 3D backbone is utilized for feature extraction, while for multi-view image data, a 2D backbone is employed to extract image features, which are subsequently transformed into a 3D voxel representation. Following the approach of MAE[19], a masking strategy is employed to input data to effectively learn representations. The voxel features are then converted to signed distance function (SDF) value and color value. By integrating predicted colors and sampled depth along rays, images and depth maps are rendered and supervised by groundtruth. In order to reduce memory costs, depth-aware ray sampling is introduced to sample only rays within a depth threshold, thus disregarding distant background.

IV. 3D Reconstruction

As shown in Tab. I, we categorized 3D reconstruction into three sub-problems: dynamic scene reconstruction, surface reconstruction, and inverse rendering. We discuss them in the following parts.

1) Dynamic Scene Reconstruction: Neural Scene Graphs (NSG) [20], for the first time, proposes to reconstruct the 3D dynamic scene with neural scene graphs that are connected by the transformation matrixes. Each node is categorized by dynamic node and static node. The dynamic node can be any dynamic actor (car and pedestrian) that is noted by 3D bounding box in each time stamp. The static node is formulated as a static background. Each node is represented by category-shared MLP and learnable embedding for each instance. During ray casting, NSG first disentangles each node by doing an AABB-ray intersection algorithm [31] in the 3D box of each node, then asks MLP to process each normalized coordinate. Finally, volume rendering is conducted in a compositional manner of depth order. These scene graph representations enable actor insertion, modification, removal, and rendering from a new viewpoint, all in a unified manner.

To achieve 3D reconstruction of large-scale driving scenes, the Block-NeRF [21] approach utilizes a divide-and-conquer strategy, breaking down the entire scene into individual blocks. Each block is then represented by a specific MLP network. During training, dynamic objects such as cars and pedestrians are masked out using semantic segmentation. To fully harness the multi-pass data collection in the same location, Block-NeRF learns appearance codes similar to NeRF-W [32] in order to control the lighting and weather of the rendered images. During inference, Block-NeRF is able to generate diverse lighting effects for the same area in the rendered images.

With the LiDAR enhancement, Neural Point Light Fields [33] use LiDAR point clouds as initialization and learn a light field to reconstruct a driving scene. When conducting volume rendering, the method selects a set of K nearest points from a point cloud for each ray. It then utilizes a light field function to predict the color of the ray, taking into account the ray's direction and features aggregated through a multi-head attention module from the closest points.

Similarly, READ [34] learns a point cloud renderer to reconstruct a 3D scene. The point cloud of the scene is obtained through the matching feature points and dense construction. Then READ learns a neural renderer using a U-net-like network. DGNR [35] also leverages point clouds as the primitives of 3D representation.

To incorporate map information, MapNeRF [36] proposed a new method to enhance out-of-trajectory driving view synthesis by incorporating map priors in driving scenes into neural radiance fields. Neural Radiance Fields with LiDAR Maps [37] proposes to incorporate LiDAR point cloud prior and GAN to benefit the training of neural radiance field.

To reconstruct large scale scenes, SUDS [22] factorizes the scene into three separate data structures to efficiently encode static, dynamic, and far-field radiance fields. They use a hash grid from instant-ngp as data structures to speed up training and inference. The dynamic branch makes use of 4D spacetime input position and time to index the feature from the hash table. They also use unlabeled inputs including images, point clouds, self-supervised 2D descriptors, and 2D optical flow to learn scene flow and semantic predictions, enabling category- and object-level scene manipulation.

Without relying on ground truth 3D box or pretrained model of depth estimation and optical flow, EmerNeRF [23] learns

TABLE I: Taxonomy of NeRF Reconstruction Research

Categories	Representative Research	Features	Primitives
Dynamic Scene Reconstruction	NSG [20]	Use 3D Box to separate objects and background	MLP
	Block-NeRF [21]	Use latent embedding to control illumination	MLP
	SUDS [22]	Dynamic hash table	Hash grid
	EmerNeRF [23]	Self-supervised scene flow estimation	Hash grid
	PVG [24], DrivingGaussian [25]	Dynamic 3D Gaussian Splatting	3D GS
Surface Reconstruction	StreetSurf [26]	Multi-scale hash grid	Hash grid
	FEGR [27]	Hybrid 3D representation	Hash grid, Mesh
	DNMP [28]	Novel scene representations	Hash grid
Inverse Rendering	FEGR [27]	Compute illumination via Monte Carlo ray tracing	Hash grid, Mesh
	UrbanIR [29]	Novel scene representations	MLP
	LightSim [30]	Physically-based rendering + learnable deferred rendering	Mesh, MLP

Fig. 7: By learning a latent embedding of each pass of the same place, Block-NeRF [21] can control the lighting effect of rendered images by changing the latent embedding.

dynamic fields in a self-surprised manner. It first learns a flow field that makes a forward and backward warping into the next or previous frame, then aggregates the per-point features. To enhance the utility for semantic scene comprehension, EmerNeRF proposes to incorporate 2D foundation model features such as DINOv2 [38] feature to benefit the training of NeRF.

In another line, UC-NeRF [39] trains NeRF in an under-calibrated camera setting. They propose 1) a layer-based color correction to address color inconsistencies in the training images, 2) virtual warping to generate more viewpoint-diverse but color-consistent virtual views for color correction and 3D recovery, and 3) a spatiotemporally constrained pose refinement designed for more robust and accurate pose calibration in multi-camera systems.

2) Surface Reconstruction: FEGR [27] learns to intrinsically decompose the driving scene using a hybrid representation of the 3D scene. Given posed images, FEGR first learns an explicit mesh using a hash grid, then estimates the spatially varying materials and HDR lighting of the underlying scene

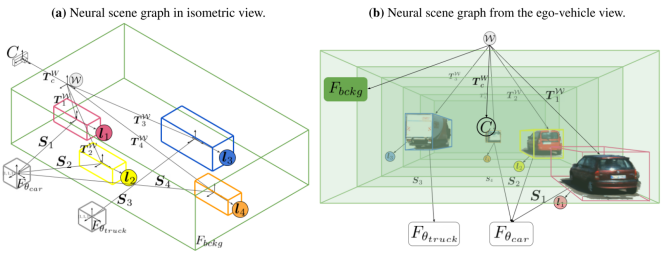

Fig. 8: The pipeline of NSG [20], which use disentangled 3D graph to represent the objects and scene.

through their proposed hybrid deferred rendering pipeline. They display satisfactory results on downstream applications such as relighting and virtual object insertion.

StreetSurf [26] develops a multi-view implicit surface reconstruction method for street view using hash tables. They disentangle the large-scale and multi-scale driving scene into three distinct parts based on their distance from the cameras: close-range, distant-view, and sky parts. For each part, they have utilized different models - a cuboid NeuS model for the close-range scene, a hyper-cuboid NeRF++ model for the distant view, and a directional MLP for the sky. Additionally, they have incorporated monocular estimated depth and normal to provide further supervision for the reconstruction process.

To further extract detailed geometric, DNMP [28] proposes to parameterize the entire scene with mesh primitives. The entire scene is voxelized and each voxel is assigned a network to parameterize the geometry and radiance of the local area. The shape of DNMP is decoded from a pretrained latent space to constrain the degree of freedom for robust shape optimization. The radiance features are associated with each mesh vertex of DNMPs for radiance information encoding.

3) Inverse Rendering: UrbanIR [29] learning to infer shape, albedo, and visibility from a single video of driving scenes. It proposes a visibility loss function, which facilitates highly accurate shadow volume estimates within the original scene. This allows for precise editing control, ultimately providing photorealistic renderings of relit scenes and seamlessly inserted objects from any viewpoint.

LightSim [30] is a neural lighting camera simulation system that enables diverse, realistic, and controllable data generation. LightSim first builds lighting-aware digital twins at scale from sensor data and decomposes the scene into dynamic actors and

static backgrounds with accurate geometry, appearance, and estimated scene lighting. Then LightSim combines physically-based and learnable deferred rendering to perform realistic relighting of modified scenes, such as altering the sun location and modifying the shadows or changing the sun brightness, producing spatially- and temporally-consistent camera videos.

4) Others: MINE [40] learns a generalizable multi-plane image feature grid for Novel View Synthesis.

PVG [24], DrivingGaussian [25] and Street Gaussians [41] use 3D Gaussian Splatting [42] to reconstruct dynamic driving scenes, displaying high-quality reconstruction and real-time rendering.

V. SLAM

Due to the powerful ability of NeRF to render images based on the pose and view orientation, the attempt to combine NeRF with pose estimation, as well as SLAM, is naturally considered and investigated by numerous researchers. Related research can be generally divided into two categories: pose estimation by NeRF and scene representation by NeRF.

A. Pose Estimation by NeRF

Several specific approaches of estimating real-time pose by NeRF have been emerged recently, and can be categoriesd into 3D implicit representation and 3D feature extraction.

1) 3D Implicit Representation: The most straightforward idea is to utilize the 3D implicit representation ability of NeRF to conduct relocalization[43–49]. Considering the pipeline of NeRF, iNeRF[43] presents an "inverting" pipeline to optimize pose estimation by a pre-trained NeRF as Fig. 9 shows. Rendered pixels are generated by NeRF from an estimated pose, which is then optimized by back-propagation of the residual between rendered and observed pixels. NeRF-Navigation[44] further combines a process loss based on the dynamic model with the photometric loss to filter the tracking results and avoid pose initialization. Besides directly comparing the observed image with the rendered one, NeRF-VINS[46] matches the observed image with the image generated by NeRF from the pose with a small offset to current estimated pose to update pose estimation. It is claimed by the authors that the synthetic image should have a significant overlapping field of view (FOV) with the observed image, which benefits the matching and pose estimation. A 2D LiDAR-based indoor Monte Carlo localization method is presented in IR-MCL[47], which predicts the occupancy probability for localization by neural network instead of volume density like NeRF. As the input and output are both light weight, IR-MCL realizes impressive real-time performance and generalizability. Considering NeRF's brilliant ability of novel view synthesis, LENS[48] applies NeRF to augment the training dataset of a learning-based pose regressor, which is then used for real-time localization. Similarly, IMA[49] trains a NeRF model conditioned on the sparse reconstruction generated by structure from motion(SfM), which is then densified by the trained NeRF to enhance relocalization.

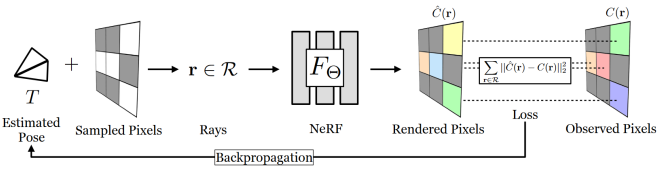

Fig. 9: The "inversion-like" pipeline of iNeRF [43]

2) 3D Feature Extraction: However, above methods all require a well-trained NeRF right in the scenario. Several researchers regard NeRF as a well-generalized 3D feature extractor for different scenarios. NeRF-Loc [50] designs a generalizable NeRF which is only conditioned on several supported images and depths to generate 3D descriptors from sampled 3D points. 2D descriptors are extracted from the query image to obtain 3D-2D correspondences and estimate relative pose by PnP (Perspective-n-Point) in a coarse-to-fine manner. In the meanwhile, Nerfels[51] also notice the 3D representation capability of NeRF. Instead of overfitting a model to the entire scene, Nerfels represents scene-agnostic local 3D patches with renderable codes, improving the generalizability. A joint PnP+photometric optimization is performed in Nerfels, resulting in improvement of wide baseline pose estimation for both hand-crafted and learned local features.

B. Scene representation by NeRF

On the contrary of optimizing pose estimation by NeRF, another application of NeRF in SLAM is representing the whole scene to optimizing mapping performance. Based on the scene representation level, we classify related research into MLP-level, voxel-level, point-level and 3D Gaussian-level representation.

1) MLP-level: The idea of optimizing mapping performance by NeRF in SLAM is initially explored in iMAP[52], which establishes parallel tracking and mapping processes sharing one MLP as the scene representation and the same loss. The tracking process optimizes the pose with respect to the fixed scene network, just similar with the pipeline of iNeRF. While in the mapping process, after keyframe selection based on information gain and active sampling guided by render loss, the whole differentiable framework can be back-propagated to jointly optimize tracking and mapping performance. iMODE[53] further realizes large-scale incremental mapping without depth input. To extract more detailed features, Li et al.[54] propose a multi-MLP neural implicit coding structure.

2) Voxel-level: Rooted from traditional MLP-based NeRF, Instant-NGP[55] encodes the scene into multi-resolution hash voxel vertices to realize real-time reconstruction, which enlights a group of NeRF-based SLAM research to represent the scene at voxel-level. The initial approach is proposed in Orbeez-SLAM[56], which applies Instant-NGP in dense mapping based on the pose estimation and keyframe selection results from classic monocular SLAM algorithm. Successive research NGEL-SLAM[57] include loop closure and global Bundle Adjustment(BA) for global pose refinement. Nevertheless, the aforementioned researches basically just inte-

grate a mature SLAM system like ORB-SLAM2[58] into the NeRF framework, exhibiting no significant intrinsic innovation within NeRF itself.

Some other research fuses NeRF with SLAM in a closer manner, optimizing pose estimation performance by a rendered voxel-level NeRF[59–64]. The most widely known research NICE-SLAM [59] incorporates multi-level local information by a hierarchical feature voxel grids, which allow updates of coarse, mid and fine level local maps, while traditional single MLP is limited by scalability. The parallel tracking and mapping processes update alternatively as in iMAP. Furthermore, Vox-Fusion[60] incrementally allocates voxels by an octree-based structure without a pre-trained geometry decoder and proposes a keyframe selection strategy suitable for sparse voxels, resulting in better performance than NICE-SLAM on the Replica dataset in both tracking and mapping. Considering the memory footprint of voxel representation, ESLAM[62] employs coarse-to-fine axis-aligned feature planes instead of feature voxel grids to reduce footprint growth by dimensionality reduction, while the divide-and-conquer scheme is leveraged in MIPS-Fusion[63] to realize scalable and robust SLAM incrementally by multi-implicit-submaps.

Another exploration of NeRF in SLAM concentrates on large-scale mapping[65–69]. Although Instant-NGP improves the reconstruction efficiency, it is still not capable for large-scale mapping because of the exponentially increasing of the hash table size. Therefore, NEWTON[65] firstly proposes a view-centric mapping approach with multiple local neural fields which are defined in local coordinate systems of each keyframe and dynamically allocated during the pose updating. Comparing with traditional world-centric neural field-based SLAM, NEWTON performs better in large-scale on-the-fly mapping. The large-scale scene is divided into multiple fix-sized cubes in [66, 67] to save computation costs. Additionally, Liu et al.[68] focus on multi-agent implicit SLAM and propose a floating-point sparse voxel octree, based on which the local map points transforming can be realized by only adjusting three vertices of the octree, therefore significantly accelerates the map fusion between multi-agents.

While many researchers concern RGB-D images as input, or directly import depth estimation results from tracking, some researchers concentrate on handling depth uncertainty of monocular SLAM in mapping[70–76]. In NeRF-SLAM [70], an uncertainty-based depth loss function is firstly designed to fully utilize SLAM outputs:

$$\mathcal{L}_D(\boldsymbol{T}, \Theta) = \|D - D^*(\boldsymbol{T}, \Theta)\|^2_{\Sigma_D}, \tag{7}$$

where D^* is the rendered depth, and D, Σ_D are the input dense depth and the depth uncertainty estimated in the tracking procedure. FMapping[72] conducts a thorough theoretical analysis to examine the depth uncertainty, and divides the uncertainty into the initialization stage and on-the-fly mapping stage, which are then managed by factorized radiance field and sliding window sampling, respectively. Another approach to cope with depth uncertainty is proposed in HI-SLAM[73] in the ray sampling procedure by sampling pixels with lower depth variance more frequently. Concentrates on the depth uncertainty of RGB-D inputs, UncLe-SLAM[76] import a Laplacian noise distribution of depth independently in each pixel.

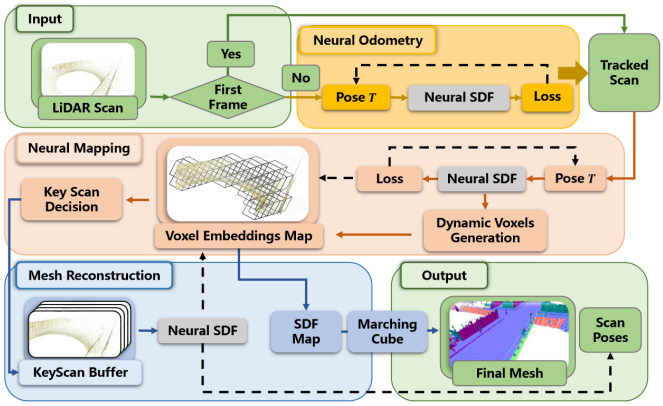

Fig. 10: The architecture of NeRF-LOAM [77]

Apart from RGB/RGB-D as inputs, plenty of researchers also engage with other modalities as inputs[77–79]. NeRF-LOAM[77] formulates a neural signed distance function(SDF) tailored for LiDAR data, which can be used to conduct tracking, mapping and key-scan selection simultaneously. The architecture is shown in Fig .10. However, due to the time-consuming intersection query between the ray and the map, NeRF-LOAM is not able to operate in real time. LONER[78] proposes a loss function derived from Jensen-Shannon(JS) Divergence to accelerate convergence while refining the real-time reconstruction performance.

Some other remarkable scene representations in NeRF-based SLAM have arisen recently. Teigen et al.[80] design a plenoxel radiance field-based SLAM system motivated by the plenoxel[81], which is an analytical radiance field representation without neural network but rather a voxel grid representation. The analytical derivative equations for tracking and mapping show both result improvement and time reduction compared with neural network-based methods. NeRF-based SLAM is explored in structural environments in Structerf-SLAM[82] by structured planar constraints.

3) Point-level: Neural point cloud-based scene representations are also anticipated to be capable for large-scale real-time tracking and mapping, as the structure of point cloud is not so compact as grids and is suitable for dynamic allocation. Point-SLAM[83] executes this strategy and dynamically adapts the anchor point density to the information density of the input RGBD image, rendering different levels of detail with different point densities, thus achieving competitive results to other dense neural RGBD SLAM methods in both tracking and mapping.

As point cloud-based representation is more light-weight and appropriate for loop closure and global pose graph optimization than MLP or voxel, CP-SLAM[84] facilitates the multi-agent SLAM system with loop closure for a single agent and cooperative localization and mapping for multiple agents with the advantage of neural point cloud. Similarly, Loopy-SLAM[85] designs point cloud submaps that grow iteratively

to perform loop closure and reduce error accumulation. PIN-SLAM[86] also incorporates point-based implicit neural representation to achieve large-scale SLAM by LiDAR.

4) 3D Gaussian-level: With the rapid development of the recent 3D Gaussian Splatting[42], plenty of 3D Gaussian-level SLAM are emerging. The first group of this type SLAM [87–90] integrate explicit 3D Gaussian representation to boost both tracking and mapping performance benefiting from the fast splatting rendering technique. Recently, SemGauss-SLAM[91] and SGS-SLAM[92] further incorporate semantic information to guide bundle adjustment and construct semantic maps for downstream tasks.

VI. Simulation

Autonomous driving simulation offers a safer and more cost-effective alternative to real-world testing by creating realistic virtual environments for sensor data generation, which facilitates the creation of diverse driving scenarios and reduces safety risks. Traditional simulation methods like CARLA [93] and AirSim [94], which rely on manual scene creation and have a significant sim-to-real gap due to handcrafted assets and simplified physics, face limitations. GeoSim [95] attempts to bridge this gap by combining graphics and neural networks for video scene generation but fails to simulate sensor data for new views. The Neural Radiance Field approach significantly enhances realism and reduces manual effort in scene creation and editing, presenting a promising solution to narrow the domain gap between the real and virtual worlds. Methods for simulation fall into two main categories: image data simulation and LiDAR data simulation. We will discuss them in the following parts.

A. Image Data Simulation

The current image data simulation methods for autonomous driving based on Neural Radiance Fields involve reconstructing scenes by using a sequence of images from real driving environments along with corresponding camera poses, allowing for the modification of vehicle behaviors within the original scenes to generate and render new photorealistic images. Depending on the representation technique, these methods are further categorized into implicit representation approaches, exemplified by NeRF, and explicit representation approaches, represented by 3D Gaussian Splatting [42].

1) Implicit Representation: These methods utilize implicit representation models similar to NeRF to reconstruct scenes. NSG[20] employs a vanilla NeRF model to represent the static background. For vehicle reconstruction, NSG reconstructs vehicles of the same category with a NeRF model, assigning each vehicle a latent code for appearance reconstruction. After training, NSG can edit the pose of vehicles in the scene by controlling their 3D bounding boxes, generating new scenes, and ultimately rendering photorealistic images. NSG transforms complex dynamic scene tasks into 3D reconstructions of multiple independent static objects by decomposing the scene into static backgrounds and vehicles using 3D bounding boxes. Limited by the capabilities of the vanilla NeRF model, NSG suffers from long training times and poor rendering

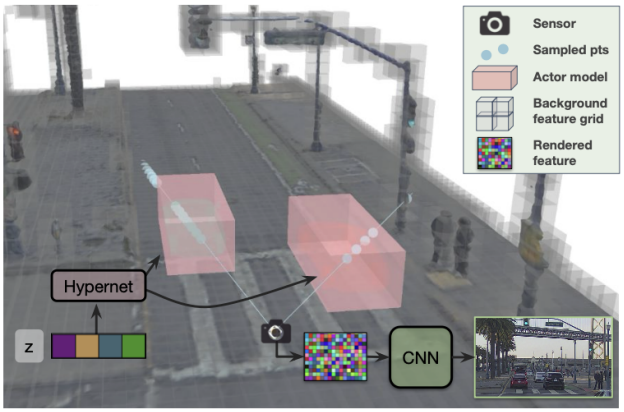

Fig. 11: Overview of UniSim[96]: First, divide the 3D scene into a static background (grey) and a set of dynamic actors (red).

quality. Instant-NGP [55] improves the efficiency of scene reconstruction and image rendering quality through the use of multilevel hash grid encoding. UniSim [96] adopts NSG's method of scene decomposition, using separate Instant-NGP models to reconstruct static backgrounds and vehicles respectively, as shown in Fig .11. To further improve the efficiency of background reconstruction, UniSim utilizes geometry priors from LiDAR observations to identify near-surface voxels and optimize only their features. When dealing with vehicles in the scene, UniSim uses a hypernetwork [97] to generate the representation of each vehicle from a learnable latent. UniSim employs Closed-loop Evaluation to demonstrate that their simulation data can be used to test the performance of autonomous vehicles in safety-critical scenarios (Figure .12).

As NeRF models are continuously optimized, the aforementioned methods are limited by the NeRF-related backbones used for scene representation. MARS[98] utilizes the framework of the NeRFStudio platform, which allows for flexible switching between different modern NeRF-related backbones and sampling strategies, to design a modular model. Moreover, besides rendering RGB images, MARS can also generate semantic segmentation images and depth maps of the scene. However, these methods require multiple neural radiance fields to represent elements in the scene, reducing the efficiency of scene reconstruction. NeuRAD [99] encodes the static background and vehicles in the scene through different multilevel hash grids and reconstructs them together using a shared neural radiance field.

2) Explicit Representation: Due to the frequent use of MLP to query information about points in a scene, their training and rendering times cannot meet real-time requirements. 3D Gaussian Splatting [42] (3DGS) can generate high-quality new viewpoint images and scene geometries while meeting real-time 3D reconstruction requirements, making simulation methods based on 3DGS for autonomous driving increasingly popular. DrivingGaussian [100] initializes the positions of 3D gaussians using LiDAR point cloud data and, similar to other methods, employs 3D bounding boxes to decompose the scene's 3D Gaussians into static backgrounds and vehicles

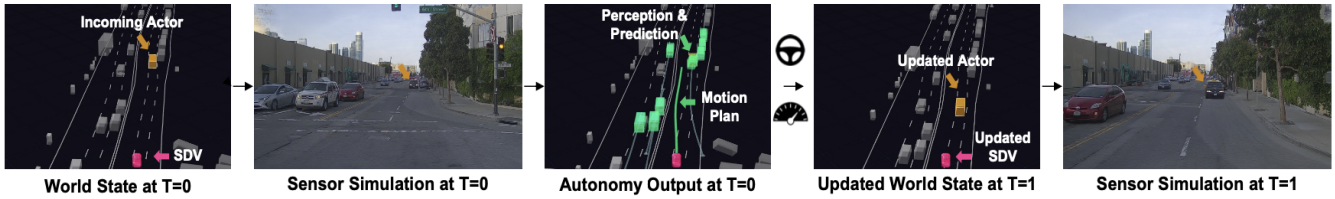

Fig. 12: UniSim Close-loop Autonomy Evaluation

within the scene, as shown in Figure 13. To apply 3DGS to large-scale static backgrounds, DrivingGaussian enhances 3DGS by introducing Incremental Static 3D Gaussians, reconstructing the complete static background by decomposing the background into multiple independent small bins and sequentially initializing the positions of 3D gaussians in each bin. DrivingGaussian uses 3D bounding boxes provided by

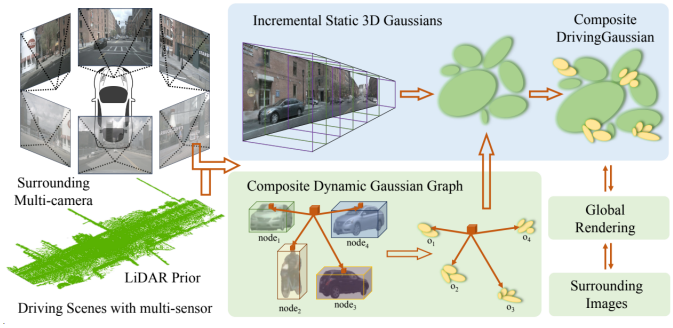

Fig. 13: Overview of DrivingGaussian.

the nuScenes dataset as the true positions of all vehicles, but in practical applications, a tracker model [101] is needed to predict the 3D bounding boxes of vehicles in images. However, tracker model-generated bounding boxes are generally noisy. Directly using them to optimize scene representation leads to a decrease in rendering quality. To address this issue, Street Gaussians [41] treats tracked poses as learnable parameters by adding a learnable transformation to each vehicle's transformation matrix. Unlike other methods based on 3DGS, Street Gaussians employ a 4D Spherical Harmonics (4D SH) model for reconstructing the color of dynamic vehicles. This allows vehicles to exhibit appearances that change over time. Street Gaussians also assigns a semantic parameter to each 3D Gaussian in space through supervised learning, aiding the model's understanding of 3D scenes. Furthermore, to achieve holistic 3D scene understanding, HUGS [102] also predicts the optical flow information of the scene, in addition to its RGB images and semantic information. Additionally, HUGS uses physical constraints derived from the unicycle model to optimize the trajectory of each vehicle's 3D bounding box, resulting in more accurate and smooth trajectories.

B. LiDAR Data Simulation

The aim of LiDAR data simulation is to utilize LiDAR measurement data to enhance neural scene representation, thus facilitating the synthesis of realistic LiDAR scans from novel viewpoints. Grounded in distinct LiDAR sensing process modeling techniques, these methodologies are mainly divided into two classifications: ray models and beam models. The following text will introduce these two methods respectively.

1) Ray Model: These methods simplify the LiDAR sensing process into a single ray, replacing the camera ray in the original NeRF model, and transform the LiDAR point cloud data into 360-degree panoramic images through spherical projection as ground truth, converting point cloud data into pseudo image data. NeRF-LiDAR [103] uses LiDAR point cloud data with semantic labels as ground truth, reconstructing 3D scenes through a neural radiance field and generating LiDAR point clouds with accurate semantic labels. To accurately reproduce the LiDAR ray dropping phenomenon, NeRF-LiDAR predicts the locations where this phenomenon occurs by training a classification mask on panoramic images. Although NeRF-LiDAR can generate LiDAR point cloud data with semantic information, it does not predict the important data of ray intensity. LiDAR-NeRF [104] also converts LiDAR point cloud data into panoramic images and produces 3D representations of the distance, the intensity, and the ray dropping probability at each pseudo-pixel. NeRF-like methods display inferior geometry in low-texture areas of large-scale scenes. To overcome this limitation, LiDAR-NeRF incorporates a structural regularization to preserve local structural details, thereby improving NeRF's ability to reconstruct geometric shapes more effectively.

2) Beam Model: Unlike the aforementioned methods, NFL [105] uses diverged beams with scattering angles to simulate the LiDAR sensing process. This technique can accurately reproduce key sensor behaviors such as beam divergence, secondary returns, and ray dropping, as shown in Figure 14.

VII. Discussion

A. Perception

In the context of the data branch, NeRF has garnered attention for its potential applications. While neural radiance fields (NeRF) have been explored for generating single-frame images, this approach proves inadequate for algorithms that depend on multi-frame inputs. Various studies, including BEVFormer[106] and the Sparse4D series[107–109], have showcased the effectiveness of integrating temporal information. The exploration of NeRF's ability to unlock the temporal-consistent data augmentation is urgently needed.

In the context of the model branch, NeRF utilizes an implicit scene representation and neural rendering to connect 3D scenes with 2D images, demonstrating advancements in

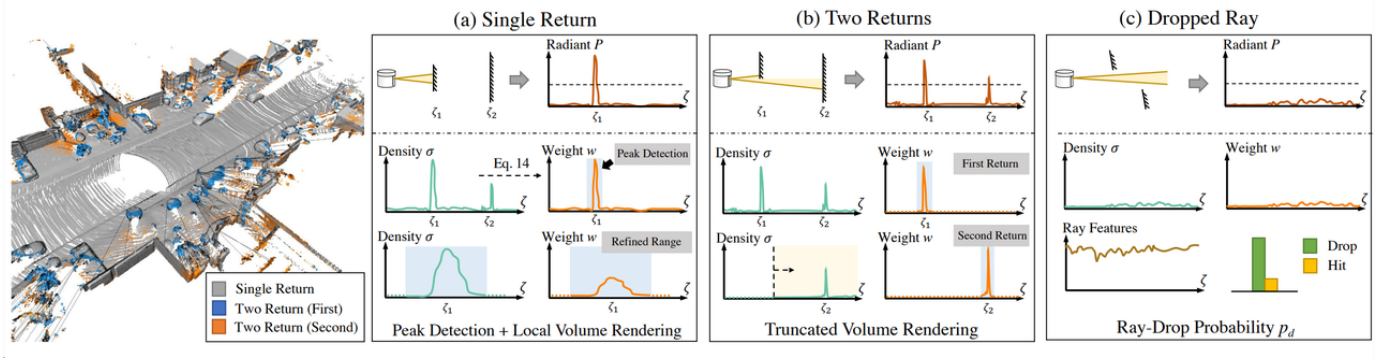

Fig. 14: The overall pipeline of NFL. (a) Single Return, (b) Two Returns, (c) Dropped Ray.

diverse perception tasks. However, the computational inefficiency of the neural rendering process poses a challenge in effectively sampling rays while upholding multi-view and temporal consistency, particularly in high-dynamic scenarios like autonomous driving.

B. Reconstruction

In the reconstruction domain, several applications have been adapted to solve industrial problems, such as transforming the data collected in one sensor setup to another sensor setup (mostly the camera intrinsic and extrinsic) to adapt to new cars, and create new data for augmentation and evaluation. However, most of the methods are limited to reconstructing rigid dynamic scenes, such as static streets with only moving vehicles, and cannot handle non-rigid dynamic objects like walking pedestrians. Future work can incorporate more object-prior to reconstruct objects. In the meantime, the reconstruction quality and runtime can still be a limitation of current methods. Future work can leverage generalizable-prior such as NeRFusion [110] to speed up reconstruction.

Furthermore, potential improvement can be leveraging the recent advances in Generative AI, generating an unlimited amount of data that is not restricted to only reconstructing real-world data. For example, researchers can use Sora from OpenAI to first generate realistic video, then reconstruct it to 3D representation, to enable diverse 3D generation.

C. SLAM

Existing NeRF-based SLAM research has the ability for autonomous driving localization and mapping. Moreover, ground truth auto-labeling and online extrinsic calibration are two potential fields for NeRF-based SLAM research.

However, current research concentrates mostly on indoor scenarios, which, though the techniques can be referred to in autonomous driving, are still not so capable of handling outdoor large-scale scenarios. Furthermore, the dynamic characteristic in autonomous driving greatly impact traditional NeRF-based SLAM research as they are tend to suffer from time-varying scenarios. To enable NeRF-based SLAM for autonomous driving, a more light-weight data structure for mapping in large-scale scenarios is in urgent need. Besides, strategies to reduce the influence of dynamic objects are also a necessity.

Another non-negligible factor is the light condition. In autonomous driving, a great number of scenarios contain severe light condition such as night or abnormal weathers like snow and fog. How to improve the robustness of NeRF-based SLAM in these scenarios presents a huge challenge. One possible solution is to introduce robust sensors such as radars to serve as a supplement.

D. Simulation

Current simulation techniques based on Neural Radiance Fields rely on a multitude of images captured from multiple perspectives to achieve more accurate geometric restoration when reconstructing vast urban scenes. Neither NeRF nor 3D Gaussian Splatting techniques can precisely restore scenes within visual blind spots, primarily due to the insufficient generalization capability of these models during scene reconstruction, failing to recover complete overall scenes from limited sparse viewpoint data. Therefore, future work will require methods based on few-shot view synthesis to address the accurate reconstruction of scenes with limited views.

Secondly, the lack of realistic interactive feedback between vehicle appearances and scene lighting leads to compromised authenticity of rendered images. Existing methods reconstruct vehicles and scenes as independent elements, thereby ignoring the impact of scene lighting on vehicle appearance. In the future, appearance editing for objects could be integrated with traditional computer graphics shading algorithms.

Moreover, objects reconstructed by existing methods are rigid, their geometric shapes do not change over time, making it a challenge to reconstruct and edit deformable objects like pedestrians while reconstructing the entire scene. In the next phase of research, it might be possible to combine existing deformable human model reconstruction methods to achieve reconstruction and editing capabilities for pedestrians.

VIII. Conclusion

In this survey, we give a comprehensive review of neural radiance field in the context of Autonomous Driving (AD). To be specific, we first introduce the basic principles and background of NeRF, then delve into a comprehensive analysis of the application of NeRF in various fields of AD, categorized as Perception, 3D Reconstruction, SLAM, and Simulation. At

last, we discuss the remaining challenges in each category and provide possible solutions. We hope this survey will facilitate future research work and promote the arrival of the era of autonomous driving.

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," 2020.

[2] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 165–174, 1984.

[3] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International Journal of Computer Vision*, vol. 126, pp. 961–972, 2018.

[4] Y. Cabon, N. Murray, and M. Humenberger, "Virtual kitti 2," *arXiv preprint arXiv:2001.10773*, 2020.

[5] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4340–4349.

[6] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 102–118.

[7] W. Tong, J. Xie, T. Li, H. Deng, X. Geng, R. Zhou, D. Yang, B. Dai, L. Lu, and H. Li, "3d data augmentation for driving scenes on camera," *arXiv preprint arXiv:2303.10340*, 2023.

[8] L. Li, Q. Lian, L. Wang, N. Ma, and Y.-C. Chen, "Lift3d: Synthesize 3d training data by lifting 2d gan to 3d generative radiance field," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 332–341.

[9] L. Li, Q. Lian, and Y.-C. Chen, "Adv3d: Generating 3d adversarial examples in driving scenarios with nerf," *arXiv preprint arXiv:2309.01351*, 2023.

[10] F. Wimbauer, N. Yang, C. Rupprecht, and D. Cremers, "Behind the scenes: Density fields for single view reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9076–9086.

[11] A. Hayler, F. Wimbauer, D. Muhle, C. Rupprecht, and D. Cremers, "S4c: Self-supervised semantic scene completion with neural fields," *arXiv preprint arXiv:2310.07522*, 2023.

[12] W. Gan, N. Mo, H. Xu, and N. Yokoya, "A simple attempt for 3d occupancy estimation in autonomous driving," *arXiv preprint arXiv:2303.10076*, 2023.

[13] M. Pan, L. Liu, J. Liu, P. Huang, L. Wang, S. Zhang, S. Xu, Z. Lai, and K. Yang, "Uniocc: Unifying vision-centric 3d occupancy prediction with geometric and semantic rendering," *arXiv preprint arXiv:2306.09117*, 2023.

[14] M. Pan, J. Liu, R. Zhang, P. Huang, X. Li, L. Liu, and S. Zhang, "Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision," *arXiv preprint arXiv:2309.09502*, 2023.

[15] J. Xu, L. Peng, H. Cheng, H. Li, W. Qian, K. Li, W. Wang, and D. Cai, "Mononerd: Nerf-like representations for monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6814–6824.

[16] Z. Xie, Z. Pang, and Y.-X. Wang, "Mv-map: Offboard hd-map generation with multi-view consistency," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8658–8668.

[17] H. Yang, H. Wang, D. Dai, and L. Wang, "Pred: Pre-training via semantic rendering on lidar point clouds," *arXiv preprint arXiv:2311.04501*, 2023.

[18] H. Yang, S. Zhang, D. Huang, X. Wu, H. Zhu, T. He, S. Tang, H. Zhao, Q. Qiu, B. Lin *et al.*, "Unipad: A universal pre-training paradigm for autonomous driving," *arXiv preprint arXiv:2310.08370*, 2023.

[19] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 16 000–16 009.

[20] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide, "Neural scene graphs for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2856–2865.

[21] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8248–8258.

[22] H. Turki, J. Y. Zhang, F. Ferroni, and D. Ramanan, "Suds: Scalable urban dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 375–12 385.

[23] J. Yang, B. Ivanovic, O. Litany, X. Weng, S. W. Kim, B. Li, T. Che, D. Xu, S. Fidler, M. Pavone *et al.*, "Emernerf: Emergent spatial-temporal scene decomposition via self-supervision," *arXiv preprint arXiv:2311.02077*, 2023.

[24] Y. Chen, C. Gu, J. Jiang, X. Zhu, and L. Zhang, "Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering," *arXiv preprint arXiv:2311.18561*, 2023.

[25] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, "Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes," *arXiv preprint arXiv:2312.07920*, 2023.

[26] J. Guo, N. Deng, X. Li, Y. Bai, B. Shi, C. Wang, C. Ding, D. Wang, and Y. Li, "Streetsurf: Extending multi-view implicit surface reconstruction to street views," *arXiv preprint arXiv:2306.04988*, 2023.

[27] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8370–8380.

[28] F. Lu, Y. Xu, G. Chen, H. Li, K.-Y. Lin, and C. Jiang, "Urban radiance field representation with deformable neural mesh primitives," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 465–476.

[29] Z.-H. Lin, B. Liu, Y.-T. Chen, D. Forsyth, J.-B. Huang, A. Bhattad, and S. Wang, "Urbanir: Large-scale urban scene inverse rendering from a single video," *arXiv preprint arXiv:2306.09349*, 2023.

[30] A. Pun, G. Sun, J. Wang, Y. Chen, Z. Yang, S. Manivasagam, W.-C. Ma, and R. Urtasun, "Neural lighting simulation for urban scenes," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=mcx8IGneYw

[31] A. Majercik, C. Crassin, P. Shirley, and M. McGuire, "A ray-box intersection algorithm and efficient dynamic voxel rendering," *Journal of Computer Graphics Techniques Vol*, vol. 7, no. 3, pp. 66–81, 2018.

[32] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7210–7219.

[33] J. Ost, I. Laradji, A. Newell, Y. Bahat, and F. Heide, "Neural point light fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 419–18 429.

[34] Z. Li, L. Li, and J. Zhu, "Read: Large-scale neural scene rendering for autonomous driving," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, 2023, pp. 1522–1529.

[35] Z. Li, C. Wu, L. Zhang, and J. Zhu, "Dgnr: Density-guided neural point rendering of large driving scenes," *arXiv preprint arXiv:2311.16664*, 2023.

[36] C. Wu, J. Sun, Z. Shen, and L. Zhang, "Mapnerf: Incorporating map priors into neural radiance fields for driving view simulation," *arXiv preprint arXiv:2307.14981*, 2023.

[37] M. Chang, A. Sharma, M. Kaess, and S. Lucey, "Neural radiance field with lidar maps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 914–17 923.

[38] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[39] K. Cheng, X. Long, W. Yin, J. Wang, Z. Wu, Y. Ma, K. Wang, X. Chen, and X. Chen, "Uc-nerf: Neural radiance field for under-calibrated multi-view cameras in autonomous driving," *arXiv preprint arXiv:2311.16945*, 2023.

[40] J. Li, Z. Feng, Q. She, H. Ding, C. Wang, and G. H. Lee, "Mine: Towards continuous depth mpi with nerf for novel view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 578–12 588.

[41] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, "Street gaussians for modeling dynamic urban scenes," *arXiv preprint arXiv:2401.01339*, 2024.

[42] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, 2023.

[43] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "inerf: Inverting neural radiance fields for pose estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1323–1330.

[44] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.

[45] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, "Loc-nerf: Monte carlo localization using neural radiance fields," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4018–4025.

[46] S. Katragadda, W. Lee, Y. Peng, P. Geneva, C. Chen, C. Guo, M. Li, and G. Huang, "Nerf-vins: A real-time neural radiance field map-based visual-inertial navigation system," *arXiv preprint arXiv:2309.09295*, 2023.

[47] H. Kuang, X. Chen, T. Guadagnino, N. Zimmerman, J. Behley, and C. Stachniss, "Ir-mcl: Implicit representation-based online global localization," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1627–1634, 2023.

[48] A. Moreau, N. Piasco, D. Tsishkou, B. Stanciulescu, and A. de La Fortelle, "Lens: Localization enhanced by nerf synthesis," in *Conference on Robot Learning*. PMLR, 2022, pp. 1347–1356.

[49] Y. Hou, T. Shen, T.-Y. Yang, D. DeTone, H. J. Kim, C. Sweeney, and R. Newcombe, "Implicit map augmentation for relocalization," in *European Conference on Computer Vision*. Springer, 2022, pp. 621–638.

[50] J. Liu, Q. Nie, Y. Liu, and C. Wang, "Nerf-loc: Visual localization with conditional neural radiance field," *arXiv preprint arXiv:2304.07979*, 2023.

[51] G. Avraham, J. Straub, T. Shen, T.-Y. Yang, H. Germain, C. Sweeney, V. Balntas, D. Novotny, D. DeTone, and R. Newcombe, "Nerfels: renderable neural codes for improved camera pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5061–5070.

[52] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Pro-*

ceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6229–6238.

[53] H. Matsuki, E. Sucar, T. Laidow, K. Wada, R. Scona, and A. J. Davison, "imode: Real-time incremental monocular dense mapping using neural field," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4171–4177.

[54] M. Li, J. He, Y. Wang, and H. Wang, "End-to-end rgb-d slam with multi-mlps dense neural implicit representations," *IEEE Robotics and Automation Letters*, 2023.

[55] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics*, vol. 41, no. 4, p. 1–15, Jul. 2022. [Online]. Available: http://dx.doi.org/10.1145/3528223.3530127

[56] 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 *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9400–9406.

[57] Y. Mao, X. Yu, K. Wang, Y. Wang, R. Xiong, and Y. Liao, "Ngel-slam: Neural implicit representation-based global consistent low-latency slam system," *arXiv preprint arXiv:2311.09525*, 2023.

[58] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[59] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 786–12 796.

[60] 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 *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022, pp. 499–507.

[61] H. Wang, J. Wang, and L. Agapito, "Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 293–13 302.

[62] M. M. Johari, C. Carta, and F. Fleuret, "Eslam: Efficient dense slam system based on hybrid representation of signed distance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 408–17 419.

[63] Y. Tang, J. Zhang, Z. Yu, H. Wang, and K. Xu, "Mips-fusion: Multi-implicit-submaps for scalable and robust online neural rgb-d reconstruction," *arXiv preprint arXiv:2308.08741*, 2023.

[64] S. Zhu, G. Wang, H. Blum, J. Liu, L. Song, M. Pollefeys, and H. Wang, "Sni-slam: Semantic neural implicit slam," *arXiv preprint arXiv:2311.11016*, 2023.

[65] H. Matsuki, K. Tateno, M. Niemeyer, and F. Tombari, "Newton: Neural view-centric mapping for on-the-fly large-scale slam," *arXiv preprint arXiv:2303.13654*, 2023.

[66] Y. Haghighi, S. Kumar, J. P. Thiran, and L. Van Gool, "Neural implicit dense semantic slam," *arXiv preprint arXiv:2304.14560*, 2023.

[67] B. Xiang, Y. Sun, Z. Xie, X. Yang, and Y. Wang, "Nisb-map: Scalable mapping with neural implicit spatial block," *IEEE Robotics and Automation Letters*, 2023.

[68] S. Liu and J. Zhu, "Efficient map fusion for multiple implicit slam agents," *IEEE Transactions on Intelligent Vehicles*, 2023.

[69] T. Deng, G. Shen, T. Qin, J. Wang, W. Zhao, J. Wang, D. Wang, and W. Chen, "Plgslam: Progressive neural scene represenation with local to global bundle adjustment," *arXiv preprint arXiv:2312.09866*, 2023.

[70] 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.

[71] Y. Zhang, F. Tosi, S. Mattoccia, and M. Poggi, "Go-slam: Global optimization for consistent 3d instant reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3727–3737.

[72] T. Hua, H. Bai, Z. Cao, and L. Wang, "Fmapping: Factorized efficient neural field mapping for real-time dense rgb slam," *arXiv preprint arXiv:2306.00579*, 2023.

[73] W. Zhang, T. Sun, S. Wang, Q. Cheng, and N. Haala, "Hi-slam: Monocular real-time dense mapping with hybrid implicit fields," *arXiv preprint arXiv:2310.04787*, 2023.

[74] H. Li, X. Gu, W. Yuan, L. Yang, Z. Dong, and P. Tan, "Dense rgb slam with neural implicit maps," *arXiv preprint arXiv:2301.08930*, 2023.

[75] 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," *arXiv preprint arXiv:2302.03594*, 2023.

[76] E. Sandström, K. Ta, L. Van Gool, and M. R. Oswald, "Uncle-slam: Uncertainty learning for dense neural slam," *arXiv preprint arXiv:2306.11048*, 2023.

[77] J. Deng, Q. Wu, X. Chen, S. Xia, Z. Sun, G. Liu, W. Yu, and L. Pei, "Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8218–8227.

[78] S. Isaacson, P.-C. Kung, M. Ramanagopal, R. Vasudevan, and K. A. Skinner, "Loner: Lidar only neural representations for real-time slam," *IEEE Robotics and Automation Letters*, 2023.

[79] X. Liu, Y. Li, Y. Teng, H. Bao, G. Zhang, Y. Zhang, and Z. Cui, "Multi-modal neural radiance field for monocular dense slam with a light-weight tof sensor," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1–11.

[80] A. L. Teigen, Y. Park, A. Stahl, and R. Mester, "Rgb-d mapping and tracking in a plenoxel radiance field," *arXiv preprint arXiv:2307.03404*, 2023.

[81] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht,

and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.

[82] H. Wang, Y. Cao, X. Wei, Y. Shou, L. Shen, Z. Xu, and K. Ren, "Structerf-slam: Neural implicit representation slam for structural environments," *Computers & Graphics*, p. 103893, 2024.

[83] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-slam: Dense neural point cloud-based slam," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 433–18 444.

[84] J. Hu, M. Mao, H. Bao, G. Zhang, and Z. Cui, "Cp-slam: Collaborative neural point-based slam system," *arXiv preprint arXiv:2311.08013*, 2023.

[85] L. Liso, E. Sandström, V. Yugay, L. Van Gool, and M. R. Oswald, "Loopy-slam: Dense neural slam with loop closures," *arXiv preprint arXiv:2402.09944*, 2024.

[86] Y. Pan, X. Zhong, L. Wiesmann, T. Posewsky, J. Behley, and C. Stachniss, "Pin-slam: Lidar slam using a point-based implicit neural representation for achieving global map consistency," *arXiv preprint arXiv:2401.09101*, 2024.

[87] C. Yan, D. Qu, D. Wang, D. Xu, Z. Wang, B. Zhao, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," *arXiv preprint arXiv:2311.11700*, 2023.

[88] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian splatting slam," *arXiv preprint arXiv:2312.06741*, 2023.

[89] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, "Gaussian-slam: Photo-realistic dense slam with gaussian splatting," *arXiv preprint arXiv:2312.10070*, 2023.

[90] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "Splatam: Splat, track & map 3d gaussians for dense rgb-d slam," *arXiv preprint arXiv:2312.02126*, 2023.

[91] S. Zhu, R. Qin, G. Wang, J. Liu, and H. Wang, "Semgauss-slam: Dense semantic gaussian splatting slam," *arXiv preprint arXiv:2403.07494*, 2024.

[92] M. Li, S. Liu, and H. Zhou, "Sgs-slam: Semantic gaussian splatting for neural dense slam," *arXiv preprint arXiv:2402.03246*, 2024.

[93] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," 2017.

[94] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," 2017.

[95] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, and R. Urtasun, "Geosim: Realistic video simulation via geometry-aware composition for self-driving," 2021.

[96] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun, "Unisim: A neural closed-loop sensor simulator," in *CVPR*, 2023.

[97] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," 2016.

[98] Z. Wu, T. Liu, L. Luo, Z. Zhong, J. Chen, H. Xiao, C. Hou, H. Lou, Y. Chen, R. Yang, Y. Huang, X. Ye, Z. Yan, Y. Shi, Y. Liao, and H. Zhao, "Mars: An instance-aware, modular and realistic simulator for autonomous driving," *CICAI*, 2023.

[99] A. Tonderski, C. Lindström, G. Hess, W. Ljungbergh, L. Svensson, and C. Petersson, "Neurad: Neural rendering for autonomous driving," 2023.

[100] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, "Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes," 2024.

[101] H. Wu, C. Wen, W. Li, X. Li, R. Yang, and C. Wang, "Transformation-equivariant 3d object detection for autonomous driving," 2022.

[102] H. Zhou, J. Shao, L. Xu, D. Bai, W. Qiu, B. Liu, Y. Wang, A. Geiger, and Y. Liao, "Hugs: Holistic urban 3d scene understanding via gaussian splatting," 2024.

[103] J. Zhang, F. Zhang, S. Kuang, and L. Zhang, "Nerf-lidar: Generating realistic lidar point clouds with neural radiance fields," 2023.

[104] T. Tao, L. Gao, G. Wang, Y. Lao, P. Chen, Z. hengshuang, D. Hao, X. Liang, M. Salzmann, and K. Yu, "Lidar-nerf: Novel lidar view synthesis via neural radiance fields," *arXiv preprint arXiv:2304.10406*, 2023.

[105] S. Huang, Z. Gojcic, Z. Wang, F. Williams, Y. Kasten, S. Fidler, K. Schindler, and O. Litany, "Neural lidar fields for novel view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 236–18 246.

[106] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *European conference on computer vision*. Springer, 2022, pp. 1–18.

[107] X. Lin, T. Lin, Z. Pei, L. Huang, and Z. Su, "Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion," *arXiv preprint arXiv:2211.10581*, 2022.

[108] ——, "Sparse4d v2: Recurrent temporal fusion with sparse model," *arXiv preprint arXiv:2305.14018*, 2023.

[109] X. Lin, Z. Pei, T. Lin, L. Huang, and Z. Su, "Sparse4d v3: Advancing end-to-end 3d detection and tracking," *arXiv preprint arXiv:2311.11722*, 2023.

[110] X. Zhang, S. Bi, K. Sunkavalli, H. Su, and Z. Xu, "NeRFusion: Fusing Radiance Fields for Large-Scale Scene Reconstruction," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Lei He (helei2023@tsinghua.edu.cn) received his B.S. in Beijing University of Aeronautics and Astronautics, China, in 2013, and the Ph.D. in the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, in 2018. From then to 2021, Dr. He served as a postdoctoral fellow in the Department of Automation, Tsinghua University, Beijing, China. He worked as the research leader of the Autonomous Driving algorithm at Baidu and NIO from 2018 to 2023. He is a Research Scientist in automotive engineering with Tsinghua University. His research interests include Perception, SLAM, Planning, and Control.

Leheng Li (lli181@connect.hkust-gz.edu.cn) earned his bachelor's degree in mathematics from the School of Mathematical Sciences, Dalian University of Technology, Dalian, China, in 2022. He is currently a Ph.D student in Artificial Intelligence at Information Hub from The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 510000, China. His research insterests are computer vision and autonomous driving.

Wenchao Sun (swc21@mails.tsinghua.edu.cn) received the B.E. degree from Tsinghua University, Beijing, China, in 2021. He is currently a Ph.D. student in mechanical engineering with the School of Vehicle and Mobility, Tsinghua University. His research interests include end-to-end autonomous driving and simulation.

Zeyu Han (hanzy21@mails.tsinghua.edu.cn) received the bachelor's degree in automotive engineering from School of Vehicle and Mobility, Tsinghua University, Beijing, China, in 2021. He is currently pursuing the Ph.D. degree in mechanical engineering with School of Vehicle and Mobility, Tsinghua University, Beijing, China. His research interests includes autonomous driving SLAM and environment understanding by 4D mmWave radars.

Yichen Liu (nz23750@bristol.ac.uk) received his bachelor of engineering degree in Telecommunications Engineering with Management from Beijing University of Posts and Telecommunications, Beijing, in 2023. He is currently pursuing a Master's degree in Robotics at the University of Bristol, Bristol, UK. His research interests include computer vision and autonomous driving.

Sifa Zheng (zsf@tsinghua.edu.cn) received the B.E. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1993 and 1997, respectively. He is currently a professor in the School of Vehicle and Mobility, and the State Key Laboratory of Automotive Safety and Energy, Tsinghua University. He is also the deputy director, Suzhou Automotive Research Institute, Tsinghua University. His current research interests include autonomous driving, vehicle dynamics and control.

Jianqiang Wang (wjqlws@tsinghua.edu.cn) received the B. Tech. and M.S. degrees from Jilin University of Technology, Changchun, China, in 1994 and 1997, respectively, and Ph.D. degree from Jilin University, Changchun, in 2002. He is currently a Professor of School of Vehicle and Mobility, Tsinghua University, Beijing, China. He has authored over 150 papers and is a co-inventor of over 140 patent applications. He was involved in over 10 sponsored projects. His active research interests include intelligent vehicles, driving assistance systems, and driver behavior. He was a recipient of the Best Paper Award in the 2014 IEEE Intelligent Vehicle Symposium, the Best Paper Award in the 14th ITS Asia Pacific Forum, the Best Paper Award in 2017 IEEE Intelligent Vehicle Symposium, the Changjiang Scholar Program Professor in 2017, Distinguished Young Scientists of NSF China in 2016, and New Century Excellent Talents in 2008.

Keqiang Li (likq@tsinghua.edu.cn) received the B.Tech. degree from Tsinghua University of China, Beijing, China, in 1985, and the M.S. and Ph.D. degrees in mechanical engineering from the Chongqing University of China, Chongqing, China, in 1988 and 1995, respectively. He is currently a Professor with the School of Vehicle and Mobility, Tsinghua University. His main research areas include automotive control system, driver assistance system, and networked dynamics and control, and is leading the national key project on CAVs (Connected and Automated Vehicles) in China. Dr. Li has authored more than 200 papers and is co-inventors of over 80 patents in China and Japan. Dr. Li has served as a Fellow Member of Chinese Academy of Engineering, a Fellow Member of Society of Automotive Engineers of China, editorial boards of the International Journal of Vehicle Autonomous Systems, Chairperson of Expert Committee of the China Industrial Technology Innovation Strategic Alliance for CAVs (CACAV), and CTO of China CAV Research Institute Company Ltd. (CCAV). He has been a recipient of Changjiang Scholar Program Professor, National Award for Technological Invention in China, etc.