# PLGSLAM: Progressive Neural Scene Represenation with Local to Global Bundle Adjustment

Tianchen Deng[1], Guole Shen[1], Tong Qin[1], Jianyu Wang[1], Wentao Zhao[1],
Jingchuan Wang[1], Danwei Wang[2], Weidong Chen[1]
[1] Shanghai Jiao Tong University [2] Nanyang Technological University


ESLAM Co-SLAM Ours

Figure 1. Large-scale indoor scene 3D Reconstruction with different methods. We depict the final mesh and camera tracking trajectory error (Absolute Trajectory Error) of different methods. The color bar on the right shows the relative scaling of color. PLGSLAM outperforms others in both scene reconstruction and pose estimation.

## Abstract

*Neural implicit scene representations have recently shown encouraging results in dense visual SLAM. However, existing methods produce low-quality scene reconstruction and low-accuracy localization performance when scaling up to large indoor scenes and long sequences. These limitations are mainly due to their single, global radiance field with finite capacity, which does not adapt to large scenarios. Their end-to-end pose networks are also not robust enough with the growth of cumulative errors in large scenes. To this end, we present PLGSLAM, a neural visual SLAM system which performs high-fidelity surface reconstruction and robust camera tracking in real time. To handle large-scale indoor scenes, PLGSLAM proposes a progressive scene representation method which dynamically allocates new local scene representation trained with frames within a local sliding window. This allows us to scale up to larger indoor scenes and improves robustness (even under pose drifts). In local scene representation, PLGSLAM utilizes tri-planes for local high-frequency features. We also incorporate multi-layer perceptron (MLP) networks for the low-frequency feature, smoothness, and scene completion in unobserved areas. Moreover, we propose local-to-global bundle adjustment method with a global keyframe database to address the increased pose drifts on long sequences. Experimental results demonstrate that PLGSLAM achieves state-of-the-art scene reconstruction results and tracking performance across various datasets and scenarios (both in small and large-scale indoor environments). The code will be open-sourced upon paper acceptance.*

## 1. Introduction

Visual Simultaneous Localization and Mapping (SLAM) has been a fundamental computer vision problem with wide applications such as autonomous driving, robotics, and virtual/augmented reality. Many traditional methods have been introduced in the past years, such as DTAM [14], ORB-SLAM [12, 13], and VINS [16]. They can estimate the camera pose and construct sparse point cloud maps in real-time with accurate localization performance. However, the sparse point cloud maps cannot meet the further perception needs of the robot. Recent attention has turned to learning-based methods for dense scene reconstruction. Kinectfusion [7], BAD-SLAM[17] reconstruct meaningful global 3D maps and show reasonable but limited reconstruction accuracy with deep learning networks.

Nowadays, with the proposal of Neural Radiance Fields (NeRF), many works focus on combining implicit scene representation with SLAM systems. iMAP [19] is the first work to use a single multi-layer perceptron (MLP) to represent the entire scene. NICE-SLAM [29] improves the scene representation method with feature grids. ESLAM [8] and Co-SLAM [23] further improve the scene representa-

tion methods. ESLAM uses tri-planes for better real-time performance and reconstruction accuracy. Co-SLAM uses joint coordinate and sparse parametric scene for accurate scene representation. They can achieve promising reconstruction quality in a small indoor room.

Although ESLAM and Co-SLAM perform well in smaller indoor scenes, they face challenges in representing large-scale indoor scenes (e.g., multi-room apartments). We outline the key challenges for real-time incremental NeRF-SLAM: ***a) insufficient scene representation capability:*** Existing methods employ a fixed-capacity, global model, limiting scalability to larger scenes and longer video sequences. ***b) accumulation of errors and pose drift:*** Existing works struggle with accuracy and robustness in large-scale indoor scenes due to accumulating errors.

To this end, we design our neural SLAM system for accurate scene reconstruction and robust pose estimation in large indoor scenes and long sequences. We propose a progressive scene representation method which dynamically initialize new scene representation when the camera moves to the bound of the local scene representation. The entire scene is divided into multiple local scene presentations, which can significantly improve the scene representation capacity of large indoor scenes. The robustness of our system is also increased because the mis-estimation is locally bounded.

In local scene representation, we combine the tri-planes with the multi-layer perceptron (MLP) to better represent the scene. We use tri-planes to encode the local high-frequency feature of the scene and use MLP to represent global low-frequency features with the coherence priors inherent. We bring together the benefits of both methods for accuracy, smoothness, and hole-filling in areas without observation.

Furthermore, we combine the traditional SLAM systems with end-to-end pose networks to improve pose estimation performance. We propose a local-to-global bundle adjustment (BA) method to eliminate the cumulative error which becomes significantly evident in large indoor scenes and long video sequences. So far, all neural SLAM systems only use end-to-end network and perform BA with rays sampled from a local subset of selected keyframes, resulting in inaccurate, non-robust pose estimation and significant cumulative errors in camera tracking. PLGSLAM maintains a global keyframe database and performs local-to-global neural warpping and reprojection Bundle Adjustment. The proposed Local-to-global BA method can eliminate the cumulative error with all the historical observations. In practice, PLGSLAM achieves SOTA performance in camera tracking and 3D reconstruction while maintaining real-time performance. **Overall, our contributions are shown as follows:**

- A progressive scene representation method is proposed which dynamically initiate local scene representation trained with frames within a local window. This enables scalability to large indoor scenes and long videos and significantly enhances the robustness.
- In local scene representation, We design a joint tri-planes and multi-layer perceptron scene representation method for accurate and smooth surface reconstruction. It can not only enhance the ability of scene representation, but also substantially reduce the memory growth from cubic to square.
- We integrate the traditional SLAM system with an end-to-end pose estimation network. A local-to-global bundle adjustment algorithm is proposed, which can mitigate cumulative error in large-scale indoor scenes. Our system manages the local and global keyframe database with the system operation, enabling bundle adjustment across all past observations, from local to global.

## 2. Related Work

**Dense Visual SLAM**. SLAM has been an active field for the past two decades. Traditional visual SLAM algorithms [9, 12, 16] estimate accurate camera poses and use sparse point clouds as the map representation. They use manipulated key points for tracking, mapping, relocalization, and loop closing. Dense visual SLAM approaches focus on reconstructing a dense map of a scene. DTAM [14] is one of the pioneer works that use the dense map and view-centric scene representation. KinectFusion [7] performs camera tracking via projective iterative-closest-point (ICP) and explicitly represents the surface of the environment via TSDF-Fusion. Some works [5, 17, 21] propose bundle adjustment(BA) method to optimize keyframe poses and construct the dense 3D structure jointly. In contrast to previous SLAM approaches, we adopt implicit scene representation of the geometry and directly optimize them during mapping.

**Implicit Scene Representation.** With the proposal of Neural radiance fields (NeRF) [11], many researchers explore taking the advantages of the implicit method into 3D reconstruction. NeRF is a ground-breaking method for novel view synthesis using differentiable rendering. However, the representation of volume densities can not commit the geometric consistency, leading to poor surface prediction for reconstruction tasks. In order to deal with it, UNISURF [15] and NeuS [24] are proposed, combining world-centric 3D geometry representation with neural radiance fields. They replace the volume density with Signed Distance Field (SDF) values. Other methods [1–3, 20] use various scene geometry representation methods, such as truncated signed distance function, voxel grid, or occupancy grid with latent codes. For large-scale representation, Mega-NeRF and LocalRF [10, 22] use multiple local scene representations for the entire scene.

**NeRF-based SLAM.** Some works focus on pose estimation of NeRF, iNeRF [28], NeRF- - [25] estimates the cam-

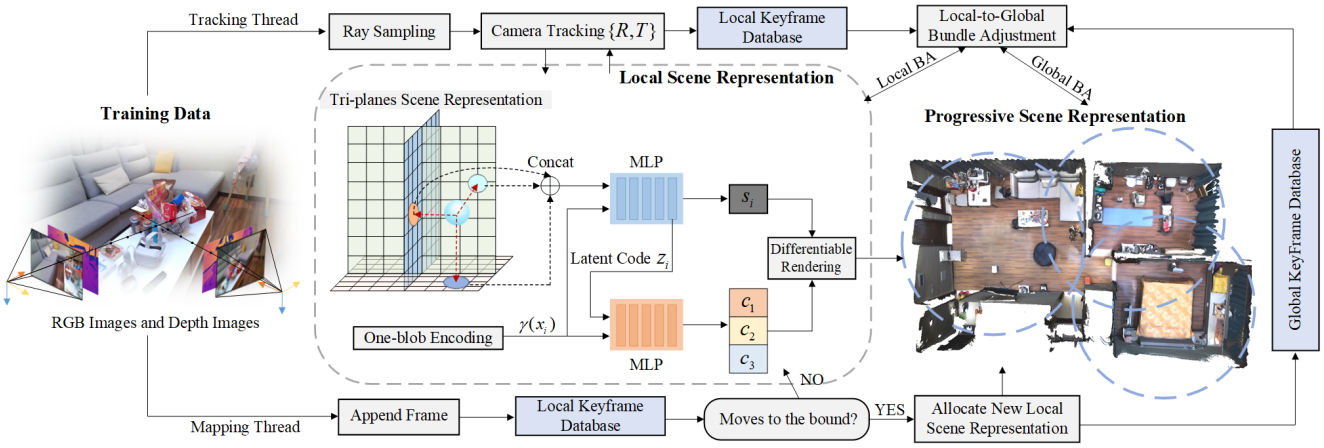


Figure 2. The isometric view of the proposed PLGSLAM system. Our system has two parallel threads: the mapping thread and the tracking thread. In the mapping thread, we propose the progressive scene representation method for the entire scene. In local scene representation, we combine the tri-planes with the multi-layer perceptron to improve the accuracy and smoothness. Both of them are online updated by minimizing our carefully designed loss through differentiable rendering with the system operating. As for the tracking thread, we propose a local-to-global bundle adjustment for accurate and robust pose estimation. Those two threads are running with an alternating optimization.

era pose with inverse NeRF optimization when the neural implicit network is fully trained. However, those methods can not optimize poses and neural implicit network simultaneously. iMAP [19] and NICE-SLAM [29] are successively proposed to combine neural implicit mapping with SLAM. iMAP uses a single multi-layer perceptron (MLP) to represent the scene, and NICE-SLAM uses a learnable hierarchical feature grid. Co-SLAM [23] and ESLAM [8] further improves the scene representation method. These are the works most relevant to our approach. However, all of them have difficult in large-scale indoor environments and long sequences. With the proposed progressive scene representation method, we can scale up to larger indoor scenarios. The fusion of tri-planes and MLP leads to high-fidelity and smooth surface reconstruction in local scene. A local-to-global bundle adjustment method is also proposed. This method can effectively eliminate growing cumulative errors in existing methods in large indoor scenes.

## 3. Method

The pipeline of our system is shown in Fig. 2. We use a set of sequential RGB-D frames $\{I_i, D_i\}_{i=1}^{M}$ with known camera intrinsic $K \in R_{3\times3}$ as our input. Our model predicts camera poses $\{R_i|t_i\}_{i=1}^{M}$, color $\mathbf{c}$, and an implicit truncated signed distance (TSDF) representation $\phi_{\boldsymbol{g}}$ that can be used in marching cubes algorithm to extract 3D meshes. For the implicit mapping thread, a progressive scene representation method (Sec. 3.1) is designed to represent large-scale indoor environments. Then, in the local radiance fields, we improve the scene representation methods and combine the tri-planes with multi-layer perceptron (MLP) by our designed architecture. Sec. 3.2 walks through the rendering process, which converts raw representations into pixel depths, colors, and sdf values. For the camera tracking thread, a local-to-global bundle adjustment method (Sec. 3.3) is designed for robust and accurate pose estimation. Several carefully designed loss functions are proposed to jointly optimize the scene implicit representation and camera pose estimation. The network is incrementally updated with the system operation.

### 3.1. Progressive Scene Representation

All the existing NeRF-based SLAM systems have difficulties in large-scale indoor scenes. They use a single, global representation of the entire environment, which limits their scene representation capacity. There are two key limitations when modeling large-scale indoor scenes: ***a)*** *the incapacity of a single, fixed-capacity model to represent videos of arbitrary length.* ***b)*** *the single scene representation tends to overfit to the early data in the sequence, leading to poorer performance in learning from the later data.* ***c)*** *any misestimation (e.g. outlier pose) has a global impact and might cause the false reconstruction.*

Mega-NeRF and Bungee-NeRF [22, 26] pre-partition the space for radiance fields. However, this approach is not applicable in our setting, as the camera poses in our system are concurrently optimized alongside the mapping thread.

In our method, we dynamically create local scene representation. Whenever the estimated camera pose trajectory leaves the space of the current scene representation, we dynamically allocate new local scene representation trained with a small set of frames. and from there, we progressively introduce subsequent local frames to the optimization. So, the entire scene can be represented as multiple local scene representations:

$$\{I_i, D_i\}_{i=1}^{M} \mapsto \{\mathrm{SR}_{\theta_1}^{1}, \mathrm{SR}_{\theta_2}^{2}, \ldots, \mathrm{SR}_{\theta_n}^{n}\} \mapsto \{\mathbf{c}, \sigma\} \quad (1)$$

where $SR_{\theta_n}^{n}$ denotes the local scene representation. Each

local scene representation is centered at the position of the last estimated camera pose. We supervise each scene representation with a local subset of the frames. Each subset contains some overlap frames, which is important for achieving consistent reconstructions in the local scene representation. Whenever the estimated camera pose leaves the bound of the current scene representation, we stop optimizing previous ones (freeze the network parameters). At this point, we can reduce memory requirements by removing unnecessary supervisory frames. We also stop updating the mapping parameters in the tracking thread to reduce errors. If the estimated camera pose is outside the current bounds, but within a previous local scene representation, we activate the previous one and proceed with the optimization process. We further increase the global consistency by inverse distance weight (IDW) fusion for all overlapping scene representations at any supervising frame.

**Local Scene Representation.** Voxel grid-based architectures [6, 23, 29] are the mainstream in NeRF-based SLAM system. However, they struggle with cubical memory growing and real-time performance. Inspired by [8], we design a tri-plane architecture joint with multi-layer perceptron (MLP). We store and optimize high-frequency features(e.g. texture) on perpendicular axis-aligned planes. The MLPs are used to encode and store low-frequency features for the coherence and smoothness priors. This joint scene representation architecture achieve high-fidelity and smoothness scene reconstruction with the ability of hole filling.

Specifically, the tri-feature planes are at two scales, i.e., coarse and fine. Coarse-level representation allows efficient reconstruction of free space with fewer sample points and optimization iterations. Then the tri-planes feature $\boldsymbol{T}(x)$ can be formulated as:

$$\begin{aligned} t^c(x) &= T^c_{xy}(x) + T^c_{xz}(x) + T^c_{yz}(x) \\ t^f(x) &= T^f_{xy}(x) + T^f_{xz}(x) + T^f_{yz}(x) \\ \boldsymbol{T}(x) &= Concat\left(t^c(x); t^f(x)\right) \end{aligned} \tag{2}$$

where $t^c(x), t^f(x)$ denote the coarse and fine feature form tri-planes. $x$ is the world coordinate. $\{T^c_{xy}, T^c_{xz}, T^c_{yz}\}$ represent the three coarse geometry feature planes, and $\{T^f_{xy}, T^f_{xz}, T^f_{yz}\}$ represent the three fine geometry feature planes.

For a sample point $x$, we use bilinearly interpolating the nearest neighbors on each feature plane. Then, we sum the interpolated coarse features and the fine, respectively, into the coarse output and fine output. At last, we concatenate the outputs together as the tri-plane features. The geometry decoder outputs the predicted SDF value $\phi_{\boldsymbol{g}}(x)$ and a feature vector $\mathbf{z}$:

$$f_g\left(\gamma(x), \boldsymbol{T}(x)\right) \rightarrow \left(\mathbf{z}, \phi_{\boldsymbol{g}}(x)\right) \tag{3}$$

where $\gamma(\mathbf{x})$ represents coordinate position encoding. We use one-blob position encoding [23] instead of embedding spatial coordinates into multiple frequency bands. Finally, the color decoder predicts the RGB value:

$$f_c(\gamma(x), \mathbf{z}, \boldsymbol{a}(x)) \mapsto \phi_{\boldsymbol{a}}(x) \tag{4}$$

$\phi_{\boldsymbol{a}}(x)$ represents the color of the sample points. Combining the MLP with the tri-planes scene representation, our architecture achieve accurate and smooth surface reconstruction, efficient memory use, and hole filling performance.

### 3.2. Differentiable Rendering

Inspired by the recent success of volume rendering in NeRF [11], we also propose to use a differentiable rendering process to integrate the predicted density and colors from our scene representation. We determine a ray $r(t) = \mathbf{o} + t\mathbf{d}$ whose origin is at the camera center of projection $o$, ray direction $r$. We uniformly sample K points. The sample bound is within the near and far planes $t_k \in [t_n, t_f]$, $k \in \{1, \ldots, K\}$ with depth values $\{\mathbf{d_1}, \ldots, \mathbf{d_K}\}$ and predicted colors $\{\mathbf{c_1}, ..., \mathbf{c_K}\}$. For all sample points along rays, we query TSDF $\phi_{\boldsymbol{g}}(p_k)$ and raw color $\phi_{\boldsymbol{a}}(p_k)$ from our networks and use the SDF-Based rendering approach to convert SDF values to volume densities:

$$\boldsymbol{\sigma}\left(x_k\right) = \frac{1}{\beta} \cdot \text{Sigmoid}\left(\frac{-\phi_{\boldsymbol{g}}\left(x_k\right)}{\beta}\right) \tag{5}$$

where $\beta \in \mathbb{R}$ is a learnable parameter that controls the sharpness of the surface boundary. Then we define the termination probability, depth, and color as:

$$\begin{aligned} w_k &= \exp\left(-\sum_{m=1}^{n-1} \boldsymbol{\sigma}\left(x_m\right)\right)\left(1 - \exp\left(-\boldsymbol{\sigma}\left(x_k\right)\right)\right) \\ \hat{\boldsymbol{c}} &= \sum_{k=1}^{N} w_k \boldsymbol{\phi_a}\left(x_k\right) \quad \text{and} \quad \hat{\boldsymbol{d}} = \sum_{k=1}^{N} w_k t_k \end{aligned} \tag{6}$$

**Depth-guided Sampling**. The depth images provide valuable geometry information which can guide neural point sampling along a ray within the bounds of depth uncertainty. we get $N$ points from stratified sampling between the near and far planes. Then, $N_{surface}$ points are drawn from near-surface points within the range $[d - \Delta d, d + \Delta d]$, where $\Delta d$ is a small offset. If the pixel does not have a valid corresponding depth value, we use an estimated depth instead.

### 3.3. Local-to-global Bundle Adjustment

Currently, existing nerf-slam methods exhibit poor accuracy and robustness in large-scale indoor scene localization. Their tracking networks are performed via minimizing rgb loss functions with respect to learnable parameters $\theta$ to estimate the relative pose matrix $\{R_i|t_i\} \in \mathbb{SE}(3)$. With the growing cumulative error $\varepsilon$ of pose estimation, those methods result in failure in large-scale indoor scenes and long

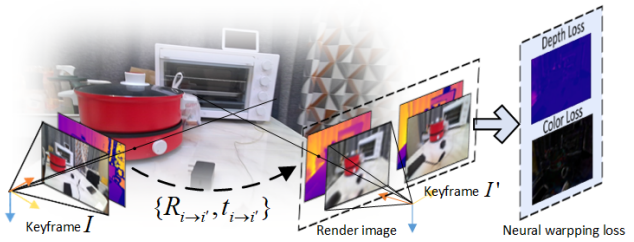


Figure 3. This figure illustrates the designed neural warping loss. We calculate the neural warpping loss between keyframe $I$ and keyframe $I'$.

videos. To this end, we design a local-to-global bundle adjustment method to solve this problem, which performs well with our progressive scene representation. We design our method by drawing inspiration from traditional keyframe-based SLAM systems for improving the robustness and accuracy of pose estimation. We propose neural warpping error and reprojection error for local-to-global bundle adjustment. The neural warpping loss is formulated as:

$$\mathcal{L}_{nwc} = \sum_{i}^{N} (\mathcal{F}(o_i, d_i, \{R_{i\to i'}, t_{i\to i'}\}) - \boldsymbol{C}_{i'})$$
$$\mathcal{L}_{nwd} = \sum_{i}^{N} (\mathcal{F}(o_i, d_i, \{R_{i\to i'}, t_{i\to i'}\}) - \boldsymbol{D}_{i'}) \quad (7)$$

Here, $L_{nwc}$ and $L_{nwd}$ are the neural warpping color and depth loss. $o_i, d_i$ denotes the rays from image $I_i$. $\{R_{i\to i'}, t_{i\to i'}\}$ denotes the relative pose from image $I_i$ to $I_{i'}$. $\mathcal{F}()$ denotes our scene representation network. We present the illustration of neural warpping loss in Fig. 3. We formulate reprojection errors with SIFT features:

$$\mathcal{L}_{re} = \sum_{i=1}^{n} \|(u_{i'}, v_{i'}) - \Pi(R_{i\to i'} P_i + t_{i\to i'})\| \quad (8)$$

where $\Pi(R_{i\to i'} P_i + t_{i\to i'})$ represents the reprojection of 3D point $P_i$ to the corresponding pixel $(u_{i'}, v_{i'})$ in image $i'$.

Whenever a keyframe arrives, we perform local bundle adjustment in our tracking and mapping thread. A keyframe is selected for every $K$ frames. When the camera moves to the bound of the current scene representation, we also initialize it as the keyframe. In local bundle adjustment, we only select keyframes from the local keyframe database that visually overlap with the current frame when optimizing the scene geometry to ensure the geometry outside the current view remains static and fast convergence. Meanwhile, we also maintain a global keyframe list with the operation of our system. After accumulating a specific number of local keyframes or the camera moves to the local bound, a global bundle adjustment is performed. In global BA, we randomly select keyframes and rays from the global keyframe database, which leverages all historical observations of the scene. This approach effectively integrates local and global information which greatly improves the robustness and accuracy of camera pose optimization in large-scale indoor scenes.

### 3.4. Objective Functions

Our mapping and tracking thread are performed via minimizing our objective functions with respect to network parameters $\theta$ and camera parameters $\{R_i|t_i\}$. The color and depth rendering losses are used in our mapping and tracking thread:

$$\mathcal{L}_c = \frac{1}{N}\sum_{i=1}^{N} (\hat{\mathbf{c}}_i - \mathbf{C_i})^2, \quad \mathcal{L}_d = \frac{1}{|R_i|}\sum_{i\in R_i} \left(\hat{\mathbf{d}}_\mathbf{i} - \mathbf{D_i}\right)^2 \quad (9)$$

where $R_i$ is the set of rays that have a valid depth observation. In addition, we design SDF loss, free space loss, and feature smoothness losses for our mapping thread. Specifically, for samples within the truncation region, we leverage the depth sensor measurement to approximate the signed distance field:

$$\mathcal{L}_{sdf} = \frac{1}{|R_i|}\sum_{r\in R_i} \frac{1}{|X_r^{tr}|}\sum_{x\in X_r^{tr}} (\phi_{\boldsymbol{g}}(x)\cdot T - (\mathbf{D_i} - \mathbf{d}))^2 \quad (10)$$

where $X_r^{tr}$ is a set of points on the ray r that lie in the truncation region, $|\mathbf{D_i} - \mathbf{d}| \le tr$. We differentiate the weights of points that are closer to the surface $X_r^{tm} = \{x | x \in |\mathbf{D_i} - \mathbf{d}| \le 0.4tr\}$ from those that are at the tail of the truncation region $X_r^{tt}$ in our sdf loss.

$$\mathcal{L}_{sdf_m} = \mathcal{L}_{sdf}\left(X_r^{tm}\right), \quad \mathcal{L}_{sdf_t} = \mathcal{L}_{sdf}\left(X_r^{tt}\right) \quad (11)$$

For sample points that are far from the surface $|D_i - d| \ge T$:

$$\mathcal{L}_{fs} = \frac{1}{|R_i|}\sum_{r\in R_i} \frac{1}{\left|X_r^{fs}\right|}\sum_{x\in X_r^{fs}} (\phi_{\boldsymbol{g}}(x) - 1)^2 \quad (12)$$

This loss can force the SDF prediction value to be the truncated distance tr. In addition, we propose feature smoothness losses to prevent the noisy reconstructions caused by tri-planes in unobserved free-space regions:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{|\mathcal{M}|}\sum_{\mathbf{x}\in\mathcal{M}} \Delta_{xy}^2 + \Delta_{xz}^2 + \Delta_{yz}^2 \quad (13)$$

where $\Delta_{xy} = \mathbf{T}(\mathbf{x} + \epsilon_{x,y}) - \mathbf{T}(\mathbf{x})$, $\Delta_{xz} = \mathbf{T}(\mathbf{x} + \epsilon_{x,z}) - \mathbf{T}(\mathbf{x})$, $\Delta_{yz} = \mathbf{T}(\mathbf{x} + \epsilon_{y,z}) - \mathbf{T}(\mathbf{x})$ denotes the feature-metric difference between adjacent sampled vertices on the three feature planes. $\mathcal{M}$ denotes a small random region form tri-planes. This loss can enhance the smoothness of our surface reconstruction results and it is only used in mapping thread.

## 4. Experiments

We validate that our method outperforms existing implicit representation-based methods in surface reconstruction, pose estimation, and real-time performance.

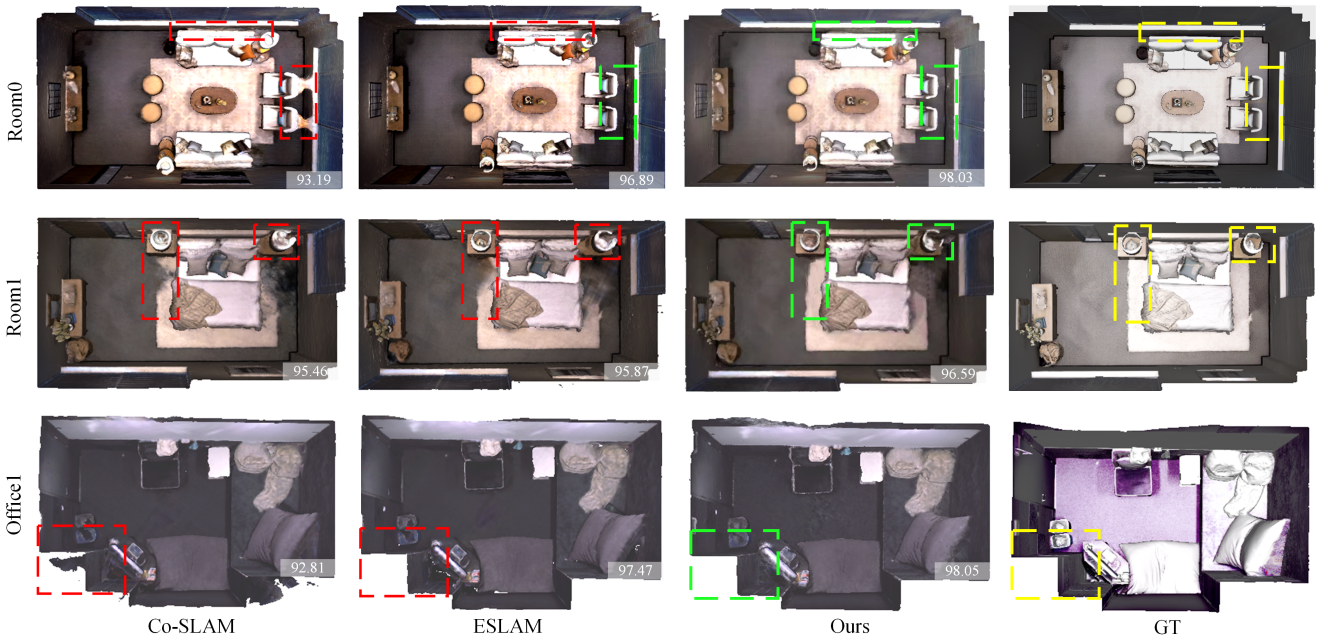


Figure 4. Reconstruction results (without cull) on Replica [18] apartment dataset. In comparison to our baselines, our methods achieve accurate and high-quality scene reconstruction and completion on various scenes.The region outlined on the image is marked in red to signify lower predictive accuracy, in green to signify higher accuracy, and in yellow to represent the ground truth results. The number in the bottom right corner of the image represents the completion ratio metric.

| Methods | Reconstruction | | | | Localization | |
|---|---|---|---|---|---|---|
| | Depth L1[cm] ↓ | Acc.[cm] ↓ | Comp.[cm] ↓ | Comp.Ratio(%) ↑ | ATE Mean[cm] ↓ | ATE RMSE[cm] ↓ |
| iMAP [19] | 4.645 | 3.624 | 4.934 | 80.515 | 3.118 | 4.153 |
| NICE-SLAM [29] | 1.903 | 2.373 | 2.645 | 91.137 | 1.795 | 2.503 |
| Vox-Fusion [27] | 2.913 | 1.882 | 2.563 | 90.936 | 1.067 | 1.453 |
| ESLAM [8] | 0.945 | 2.082 | 1.754 | 96.427 | 0.565 | 0.707 |
| Co-SLAM [23] | 1.513 | 2.104 | 2.082 | 93.435 | 0.935 | 1.059 |
| Ours | **0.771** | **1.793** | **1.543** | **97.877** | **0.525** | **0.635** |

Table 1. Quantitative results of our proposed PLGSLAM with existing NeRF-SLAM system on the Replica dataset [18]. We evaluate reconstruction and localization performance in small room scenes. The results are the average on the scenes of the Replica dataset. Our method outperforms the existing method in surface reconstruction and pose estimation.

| Methods | Reconstruction[cm] | | | Localization[cm] | |
|---|---|---|---|---|---|
| | Acc. | Comp. | Comp.Ratio(%) | Mean | RMSE |
| NICE-SLAM[29] | 29.17 | 4.45 | 67.97 | 8.78 | 9.63 |
| ESLAM[8] | 26.22 | 4.53 | 71.43 | 7.89 | 8.95 |
| Co-SLAM[23] | 26.55 | 4.67 | 70.34 | 7.67 | 8.75 |
| Ours | **19.42** | **4.21** | **74.48** | **6.12** | **6.77** |

Table 2. Camera tracking results on the Scannet datasets [4]. We evaluate our camera tracking performance on the Scannet dataset to verify the effectiveness of our method. Our method achieves high-fidelity surface reconstructions and superior camera tracking.

### 4.1. Datasets and Metrics

**Datasets.** We evaluate PLGSLAM on a variety of scenes from different datasets. We quantitatively evaluate the reconstruction quality on 8 small room scenes from Replica [18] (**nearly** $6.5m \times 4.2m \times 2.7m$ with 2000 images). We evaluate on real-world scenes from ScanNet [4] for long sequences (more than 5000 images) and large-scale indoor scenarios (**nearly** $7.5m \times 6.6m \times 3.5m$). We also evaluate on Apartment dataset of the multi-rooms scene (**nearly** $14.5m \times 7.5m \times 3.8m$ with more than 12000 images) from NICE-SLAM [29].

**Metrics.** We use Depth L1 (cm), Accuracy (cm), Completion (cm), and Completion ratio $(\%)$ to evaluate the reconstruction quality. Following NICE-SLAM and ESLAM[8, 29], we perform frustum and occlusion mesh culling that removes unobserved regions outside frustum and the noisy points within the camera frustum but outside the target scene. However, this simple strategy removes too many meshes, leading to excessive holes and ineffective assessment of the reconstruction results. For the evaluation of camera tracking, we adopt ATE RMSE and Mean(cm).

| Methods | Reconstruction | | | | Localization | |
|---|---|---|---|---|---|---|
| | Depth L1[cm] ↓ | Acc.[cm] ↓ | Comp.[cm] ↓ | Comp.Ratio(%) ↑ | ATE Mean[cm] ↓ | ATE RMSE[cm] ↓ |
| iMAP [19] | 24.558 | 14.296 | 7.476 | 44.422 | 9.963 | 10.612 |
| NICE-SLAM [29] | 37.052 | 6.064 | 5.576 | 71.792 | 4.776 | 5.394 |
| Vox-Fusion [27] | 43.077 | 26.375 | 9.454 | 49.554 | 11.473 | 12.754 |
| ESLAM [8] | 16.355 | 17.546 | 4.301 | 71.626 | 6.637 | 7.283 |
| Co-SLAM [23] | 6.702 | 13.355 | 3.666 | 80.486 | 6.182 | 6.891 |
| Ours | **6.033** | **11.086** | **3.261** | **86.557** | **5.574** | **6.228** |

Table 3. Quantitative results of our proposed PLGSLAM with existing NeRF-SLAM system on the Apartment dataset [29]. We evaluate reconstruction and localization performance in large-scale multi-room scenes. The results are the average of three runs. Our method outperforms the existing method in surface reconstruction and pose estimation.

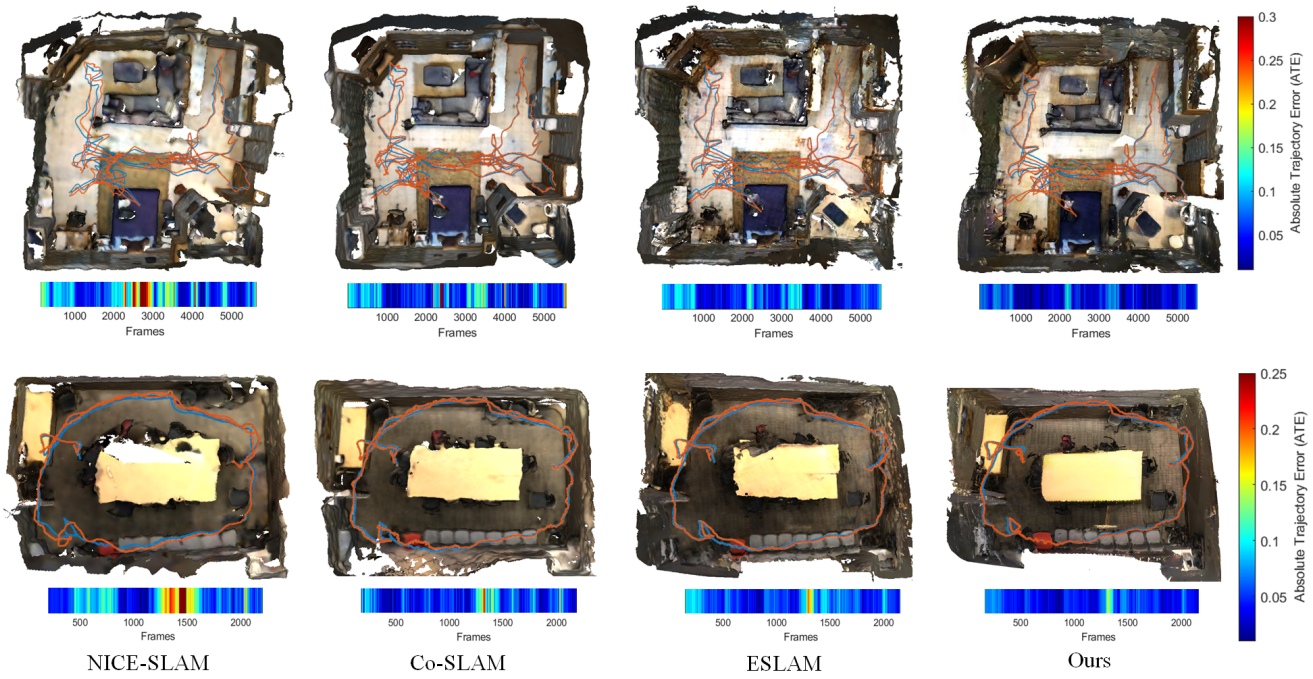


Figure 5. Qualitative comparison of our proposed PLGSLAM method's surface reconstruction and localization accuracy with existing NeRF-based dense visual SLAM methods, NICE-SLAM [29], Co-SLAM [23], and ESLAM [8] on the ScanNet dataset [4]. The ground truth camera trajectory is shown in blue, and the estimated trajectory is shown in red. Our method predicts more accurate camera trajectories and does not suffer from drifting issues. We also visualize the Absolute Trajectory Error ATE (bottom color bar) of different methods. The color bar on the right shows the relative scaling of color. It should also be noted that our method runs faster on this dataset.

**Implementation** We run PLGSLAM on a desktop PC with NVIDIA RTX 3090ti GPU. For experimental settings, the truncation distance $tr$ is set to 6 cm. We employ feature planes with a resolution of 24 cm for coarse tri-planes. We use 6cm resolution for fine tri-planes. All feature planes have 32 channels, resulting in a 64-channel concatenated feature input for the decoders. The decoders are two-layer MLPs with 32 channels in the hidden layer. For Replica [18], we sample $N$ = 32 points for stratified sampling and $N_{surface}$ = 8 points for importance sampling on each ray. And for ScanNet [4], we set $N$ = 48 and $N_{surface}$ = 8. We set $\lambda_{fs} = 5$ , $\lambda_{sdf_m} = 200$ ,$\lambda_{sdf_t} = 10$ , $\lambda_{smooth} = 0.01$, $\lambda_d = 0.1$ , and $\lambda_c = 5$ . And during tracking, we set $\lambda_{fs} = 10$ , $\lambda_{sdf_m} = 200$, $\lambda_{sdf_t} = 50$ , $\lambda_{re} = 10$ , $\lambda_{nw} = 10$ , $\lambda_d = 1$, and $\lambda_c = 5$ . For further details of our implementation, refer to the supplementary.

### 4.2. Experimental Results

**Replica dataset**. We evaluate on the same RGB-D sequences as ESLAM [8] and Co-SLAM [23]. We use this dataset to test our system performance in small room scenes (nearly $6.5m \times 4.2m \times 2.7m$). As shown in Tab. 1, our method achieves higher reconstruction and pose estimation accuracy. We show the qualitative results in Fig. 4. We can see that ESLAM maintains more reconstruction details,

| | Method | Speed FPT(s) | Memory Grow.R. |
|---|---|---|---|
| Replica[18] | NICE-SLAM[29] | 2.10 | $O(L^3)$ |
| | ESLAM[8] | 0.18 | $O(L^2)$ |
| | Co-SLAM[23] | 0.16 | $O(L^3)$ |
| | Ours | **0.12** | $O(L^2)$ |
| Scannet[4] | NICE-SLAM[29] | 3.35 | $O(L^3)$ |
| | ESLAM[8] | 0.55 | $O(L^2)$ |
| | Co-SLAM[23] | 0.38 | $O(L^3)$ |
| | Ours | **0.37** | $O(L^2)$ |

Table 4. Runtime analysis of our method in comparison with existing ones in terms of average frame processing time (AFPT), and model size growth rate w.r.t. scene side length L. We evaluate these method on replica dataset [18] and Scannet dataset [4]. Our method is greatly faster and the model size grow is significantly reduced from cubic to square.

| Methods | Reconstruction[cm] | | | Localization[cm] | |
|---|---|---|---|---|---|
| | Acc. | Comp. | Comp.Ratio(%) | Mean | RMSE |
| w/o joint enc. | 13.314 | 4.687 | 81.347 | 5.935 | 6.787 |
| w/o prog. | 12.754 | 4.231 | 83.156 | 5.875 | 6.693 |
| w/o lg BA | 12.435 | 4.181 | 83.473 | 5.874 | 6.591 |
| Ours | **11.086** | **3.261** | **86.557** | **5.574** | **6.228** |

Table 5. Ablation study. We conduct experiments on Apartment dataset [29] to verify the effectiveness of our method. Our full model achieves better completion reconstructions and more accurate pose estimation results.

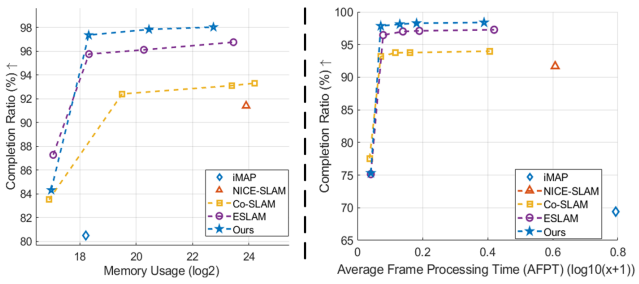


Figure 6. Completion ratio vs. model size and average time for PLGSLAM with other methods. Each model corresponds to a different hash-table size. iMAP and NICE-SLAM shown for reference.

but the results contain some artifacts. Co-SLAM achieves smooth completion in unobserved areas, but the accuracy of the reconstruction and pose estimation is relatively low. Our method successfully achieves consistent completion as well as high-fidelity reconstruction results.

**Scannet dataset.** We evaluate the camera tracking and reconstruction results of PLGSLAM on real-world large room sequences (nearly $7.5m \times 6.6m \times 2.7m$) from ScanNet [4] . We use the absolute trajectory error (ATE) as our metric. Tab. 2 shows that our method achieves better pose estimation and surface reconstruction results in comparison to NICE-SLAM [29], ESLAM [8], and Co-SLAM [23]. PLGSLAM exhibits superior scene representation capabilities and more accurate and robust tracking performance in large-scale indoor scenes. Fig. [4] also shows PLGSLAM achieves better reconstruction quality with smoother results and finer details.

**Apartment dataset.** We evaluate the surface reconstruction and camera tracking accuracy of PLGSLAM on Apartment dataset (**nearly** $14.5m \times 7.5m \times 3.2m$). Tab. 3 shows that quantitatively, our method achieves SOTA tracking results in comparison to Co-SLAM and ESLAM. These algorithms typically exhibit significant cumulative errors in large-scale indoor dataset scenarios. Fig. 1 also shows PLGSLAM achieves better reconstruction quality with smoother results and finer details.

### 4.3. Ablation Study

In this section, we conduct various experiments to verify the effectiveness of our method. Tab. 5 illustrates a quantitative evaluation with different settings.

**Joint scene representation.** It is obvious that the joint scene representation (tri-planes with MLP) significantly improves our surface reconstruction accuracy.

**Progressive scene representation.** We replace our progressive scene representation and use a single network for the entire scene. We can observe that this method has a great influence on pose estimation and reconstruction metrics. This network significantly improves the capacity of scene geometry representation and enhances the robustness for local misestimation.

**Local-to-global bundle adjustment.** We remove our local-to-global bundle adjustment in this experiment. Our full model leads to higher accuracy and better completion. The local-to-global BA can significantly reduce the growing cumulative error, which is considerably harmful in large-scale scenes. This method can greatly improve the robustness and accuracy of the camera tracking.

## 5. Conclusion

In this paper, we propose a novel dense SLAM system, PLGSLAM, which achieve accurate surface reconstruction and pose estimation in large indoor scenes. Our progressive scene representation method enables our system to represent large-scale indoor scenes and long videos. The joint encoding method with the tri-planes and multi-layer perceptron further improves the accuracy of local scene representation. The local-to-global bundle adjustment method combines the traditional SLAM method with end-to-end pose estimation, which achieves robust and accurate camera tracking and mitigate the influence of cumulative error and pose drift. Our extensive experiments demonstrate the effectiveness and accuracy of our system in both scene reconstruction, depth estimation, and pose estimation.

## 6. Acknowledgement

Authors gratefully appreciate the contribution of Yanbo Wang from Shanghai Jiao Tong University, Shenghai Yuan from Nanyang Technological University.

## References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6290–6301, June 2022. 2

[2] Aljaz Bozic, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. Transformerfusion: Monocular rgb scene reconstruction using transformers. *Advances in Neural Information Processing Systems*, 34:1403–1414, 2021.

[3] Jaesung Choe, Sunghoon Im, Francois Rameau, Minjun Kang, and In So Kweon. Volumefusion: Deep depth fusion for 3d scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16086–16095, October 2021. 2

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 6, 7, 8

[5] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph.*, 36(4), jul 2017. 2

[6] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 4

[7] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011. 1, 2

[8] Mohammad Mahdi Johari, Camilla Carta, and François 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*, pages 17408–17419, 2023. 1, 3, 4, 6, 7, 8

[9] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007. 2

[10] Andreas Meuleman, Yu-Lun Liu, Chen Gao, Jia-Bin Huang, Changil Kim, Min H Kim, and Johannes Kopf. Progressively optimized local radiance fields for robust view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16539–16548, 2023. 2

[11] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 4

[12] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1, 2

[13] Raúl Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 1

[14] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011. 1, 2

[15] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5589–5599, October 2021. 2

[16] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018. 1, 2

[17] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[18] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 6, 7, 8

[19] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *ICCV*, pages 6229–6238, October 2021. 1, 3, 6, 7

[20] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15598–15607, June 2021. 2

[21] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *ICLR*, 2018. 2

[22] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12922–12931, 2022. 2, 3

[23] Hengyi Wang, Jingwen Wang, and Lourdes 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*, pages 13293–13302, 2023. 1, 3, 4, 6, 7, 8

[24] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27171–27183. Curran Associates, Inc., 2021. 2

[25] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2

[26] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *European conference on computer vision*, pages 106–122. Springer, 2022. 3

[27] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507, 2022. 6, 7

[28] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330, 2021. 2

[29] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, pages 12786–12796, June 2022. 1, 3, 4, 6, 7, 8