City-scale Incremental Neural Mapping with Three-layer Sampling and Panoptic Representation

Yongliang Shi[1*], Runyi Yang[1,2*], Pengfei Li[1], Zirui Wu[1,2], Hao Zhao[3], Guyue Zhou[1†]

***Abstract*— Neural implicit representations are drawing a lot of attention from the robotics community recently, as they are expressive, continuous and compact. However, city-scale incremental implicit dense mapping based on sparse LiDAR input is still an under-explored challenge. To this end, we successfully build the first city-scale incremental neural mapping system with a panoptic representation that consists of both environment-level and instance-level modelling. Given a stream of sparse LiDAR point cloud, it maintains a dynamic generative model that maps 3D coordinates to signed distance field (SDF) values. To address the difficulty of representing geometric information at different levels in city-scale space, we propose a tailored three-layer sampling strategy to dynamically sample the global, local and near-surface domains. Meanwhile, to realize high fidelity mapping, category-specific prior is introduced to better model the geometric details, leading to a panoptic representation. We evaluate on the public SemanticKITTI dataset and demonstrate the significance of the newly proposed three-layer sampling strategy and panoptic representation, using both quantitative and qualitative results. Codes and data will be publicly available.**

I. Introduction

City-scale 3D maps with efficient memory and rich information could be used for more efficient and accurate state estimation and smoother path planning in autonomous driving.

There are varieties of explicit scene representations for 3D map, such as point clouds[34], voxel grids[29], octrees[10], surfel clouds[4] or polygon meshes. They directly sample points on the surface together with additional information like surface normals or point radius, or connectivity of points. They are useful for applications such as navigation and manipulation, which have led to significant progress in 3D scene understanding. However, these scene representations are discrete, limiting the achievable spatial resolution.

Contrary to explicit representations, implicitly defined, continuous, differentiable shape representations parameterized by MLP have emerged as a powerful paradigm. It is memory-efficient and easily deals with a wide variety of surface topologies without resolution limitation, enabling downstream tasks ranging from robotic perception[9] and 3D reconstruction to navigation[1]. Recently, research about RGBD-based incremental implicit mappings has made significant progress[18], [32], but they are all used for indoor scene reconstruction. Due to the fact that the point cloud of LiDAR is so sparse, it is difficult to depict geometry details. Consequently, transplanting these methods to city scenes directly will suffer from limitations: **1) Scale gap**. There is a wide gap between a scene and its instances in scale, which results in the difficulty in capturing geometric details of scene and instances simultaneously with traditional uniform sampling. **2) Lost of high frequency information**. Owing to the lack of prior knowledge and the powerful fitting ability of MLP, the implicit field is too smooth to describe the details of the object surface in the scene, leading to distortions especially at edges of object. Moreover, current works are poor in semantic segmentation.

To address these limitations, we propose an incremental mapping system based on hybrid implicit representation under city-scale scene with LiDAR (Fig.1). Unlike implicit SLAM systems that incrementally update pose estimation and scene reconstruction, we focus on SDF-based semantic mapping with a novel sampling strategy and panoptic representation. To overcome the scale gap, three-layer sampling that covers global, local and near-surface is proposed to facilitate the continual learning system that predict the SDF value of the whole scene space. Besides, for high fidelity instance reconstruction, a category-shared MLP is trained in dense shapes as geometry prior. Given partial point clouds captured by LiDAR, corresponding latent codes are optimized by the fixed MLP to complete panoptic representations of instances. And the hybrid representation makes the scene looks more realistic, as is shown in Fig.5(a). What's more, for better map interactivity, a parallel MLP with similar architecture is trained for semantic segmentation of the scene. Finally, to avoid the impact of scenes with single geometric information such as road and buildings on quantitative evaluation, we design a Class-aware Chamfer Distance (CCD), which weighted the chamfer distance according to the proportion of the number of point clouds in different categories. We verify our method on SemanticKitti's odometry, and demonstrate the effectiveness and superiority of our method.

To summarize, our contributions are as follows:

- Three-layer sampling whose parameters consistently change when new LiDAR frames are available.
- A city-scale incremental learning system for scene is developed.
- A new hybrid panoptic representation is proposed to capture both low-frequency environment elements and high-frequency instances.
- On the city-scale dataset SemanticKITTI, better performance are demonstrated when using our method.

*Equal contribution, †Corresponding author

[1]Institute for AI Industry Research (AIR), Tsinghua University, China, {shiyongliang, lipengfei, zhouguyue} @air.tsinghua.edu.cn.

[2]Beijing Institute of Technology, China, {wuzirui, runyi.yang} @bit.edu.cn.

[3]Intel Labs, China, hao.zhao@intel.com.

II. Related Work

Implicit neural representations are about to parameterize a continuous differentiable signal with a neural network, providing a possibly more compact representation of scenes and instances. There has been much promising recent work on using neural implicit representations.

Implicit Scene Representation The Scene Representation Network (SRN)[24] was one of the first methods to use a multi-layer perceptron (MLP) as the neural representation of a learned scene given a collection of images and associated poses. Occupancy Networks [20], [15] learned an implicit 3D occupancy function for shapes or large scale scenes given 3D supervision. Signed distance fields (SDFs) have some useful properties, for instance, unlike explicit representations, they allow for changes of surface topology, and they can be updated very easily. A variety of SDF-based implicit neural representations are proposed to solve boundary value problems. However, ReLU-based MLP [21], [30], [8] are incapable of reconstructing high-frequency details of surfaces because they are piecewise linear and their derivative is zero. Sitzmann leverage MLPs with periodic activation functions for implicit neural representations[23]. What's more, NeRF[16] has drawn wide attention across scene representation [17], [27], [22] due to its simplicity and extraordinary performance.

Continual Learning of Scene Yan[32] proposed SDF-based continual neural mapping to update network parameters at each time when new observations arrive that lead to a self-improved mapping function. Following the NeRF[16], iMAP[25] realizes implicit SLAM in a real sense for the first time. NICE-SLAM[36] optimizes the iMAP with pre-trained geometric priors and enables detailed reconstruction on large indoor scenes. Given a stream of posed depth images, iSDF[18] used a neural network to regresses input 3D coordinates to signed distance and complete real-time reconstruction. Azinović[2] incorporates the truncated signed distance function (TSDF) in the NeRF framework to represent surface instead of volumetric, meanwhile, pose and camera refinement technique is proposed to improve the overall reconstruction quality. All the above works are incremental implicit reconstruction of indoor scenes based on RGB-D sensors. At present, there is no work based on urban incremental implicit reconstruction of sparse LiDAR point clouds.

Panoptic Representation The aforementioned works mostly focus on learning representations for whole scene within a few categories, and they have not been studied details of instances in large scenes. Jiang[11] learn to encode/decode geometric parts of objects at a part scale by training an implicit function auto-encoder, and optimize Latent Implicit Grid representation that matches a partial scene observation. Yang[33] used a shared MLP with instance-specific latent codes to incorporate prior. Kundu[13] uses meta-learning to find a good category-specific initialization and employ instance-specific fully weight encoded functions to represent each object in scene. Most of the previous work is based on dense point clouds or images to complete instance reconstruction, Boulch[6] adopted sampling strategy of picking needles with end points on opposite sides or on the same side of the surface to realize dense reconstruction with sparse point cloud, this method requires pre-training on the dataset where sparse point clouds reside.

III. Formulation

Scene: We are committed to modelling 3D environment W incrementally through continuous LiDAR streams D^t with an implicit representation $\mathcal{F}(\cdot)$. The $D^t = (\boldsymbol{x}_i^t, \boldsymbol{s}_i^t)$ is composed of point cloud coordinates $\boldsymbol{x}_i^t$ and corresponding attributes $\boldsymbol{s}_i^t$.

$$\boldsymbol{s} = \mathcal{F}(\boldsymbol{x}; \theta) \tag{1}$$

Here, $\boldsymbol{s}$ is represented by the SDF whose sign indicates whether the region is inside (-) or outside (+) of the shape. It is a continuous function that maps spatial point $\boldsymbol{x}$ to its distance to the surface boundary, and the implicitly defined surface $\mathcal{S}$ of scene is represented by the iso-surface of $\mathcal{F}(\cdot) = 0$:

$$\mathcal{S} = \left\{ \boldsymbol{x} \in \mathbb{R}^3 \mid \mathcal{F}\left(\boldsymbol{x}; \theta^t\right) = 0 \right\}, \mathcal{F}(\cdot) : \mathbb{R}^3 \mapsto \mathbb{R}. \tag{2}$$

This problem is cast as a function that satisfies a set of M constraints satisfying M constraints $\mathcal{C}_m$, each of which relate the function $\mathcal{S}$ and its derivative $\nabla_{\boldsymbol{x}}\mathcal{S}$ to quantities $\boldsymbol{a}(\boldsymbol{x})$ on their corresponding domain Ω_m.

$$\mathcal{C}_m\left(\boldsymbol{a}(\boldsymbol{x}), \mathcal{S}, \nabla_{\boldsymbol{x}}\mathcal{S}\right) = 0, \forall \boldsymbol{x} \in \Omega_m, m = 1, \ldots, M \tag{3}$$

Formally, by optimising the weights θ, our goal is to learn a neural network $\mathcal{F}(\cdot)$ that parameterizes $\boldsymbol{s}$ to model a scene from LiDAR streams D^t incrementally to to describe the relationship between spatial coordinates and scene attributes.

Instance: The instance aware representation is to complete object reconstruction combining constraints of the same category with similar shape and appearance. Given a dataset of N training shapes $\mathcal{S}_{i=1}^N$ represented with signed distance function $f_{\theta_{i=1}}^N$:

$$\mathcal{S}_i := \{(\boldsymbol{x_j}, s_j) : f_\theta(\boldsymbol{x_j}) = s_j\} \tag{4}$$

We adopt a probabilistic perspective to derive the shared-attribute-MLP based instance reconstruction in the case of incomplete information. The posterior over shape code latent code z_i which is paired with training shape $\mathcal{S}_i$ can be decomposed as:

$$p_\theta\left(\boldsymbol{z}_i \mid \mathcal{S}_i\right) = p\left(\boldsymbol{z}_i\right) \prod_{(\boldsymbol{x}_j, \boldsymbol{s}_j) \in X_i} p_\theta\left(\boldsymbol{s}_j \mid z_i; \boldsymbol{x}_j\right) \tag{5}$$

where the SDF likelihood $p_\theta\left(\boldsymbol{s}_j \mid z_i; \boldsymbol{x}_j\right)$ is expressed via a deep feed-forward network $f_\theta\left(\boldsymbol{z}_i, \boldsymbol{x}_j\right)$, prior distribution over codes $p\left(\boldsymbol{z}_i\right)$ is assumed to be a zero-mean multivariate-Gaussian with a spherical covariance $\sigma^2 I$. At training time we maximize the joint log posterior over all training shapes with respect to the individual shape codes $z_{i=1}^N$ and the network parameters θ, making f_θ a good approximator of the given SDF.

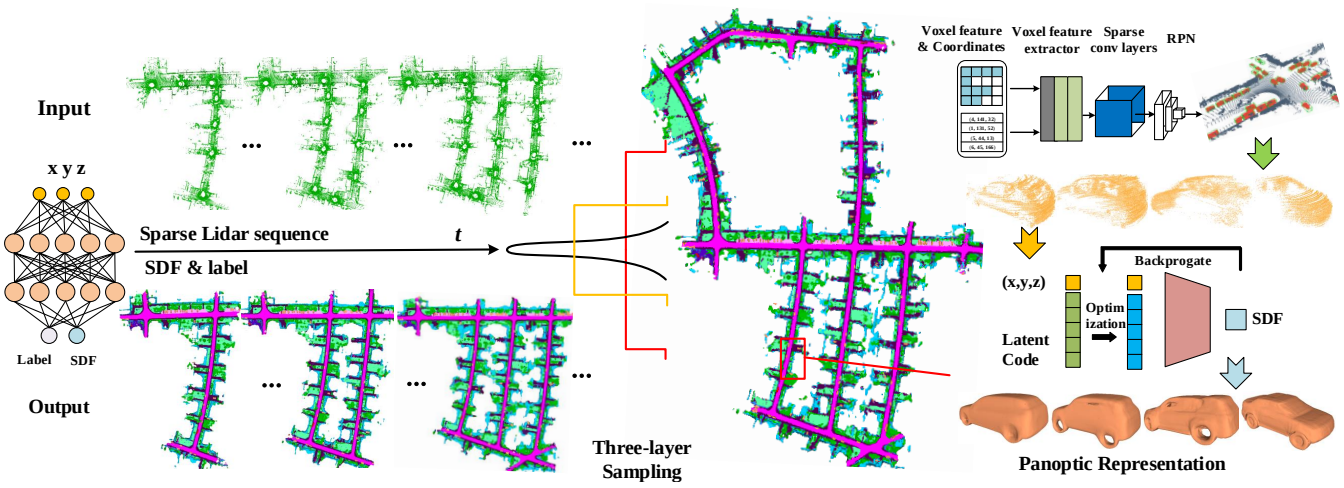


Fig. 1. Given sequential sparse data, our model continuously learns scene property with three-layer sampling strategy that covers different level information including global,local and near-surface to achieve implicit semantic scene representation. In addition, we pre-train a category-specific MLP as prior to complete panoptic representation of vehicles even with serious data default in the scene.

After training and fixing θ of category-specific MLP, a shape code $\hat{\boldsymbol{z}}$ for target shape $\mathcal{S}$ can be estimated via Maximum-a-Posterior (MAP) estimation as:

$$\hat{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} \sum_{(\boldsymbol{x}_j, \boldsymbol{s}_j) \in \mathcal{S}} \mathcal{L}\left(f_\theta\left(\boldsymbol{z}, \boldsymbol{x}_j\right), s_j\right) + \frac{1}{\sigma^2}\|\boldsymbol{z}\|_2^2 \quad (6)$$

IV. Method

A. Network architecture

Following the network architecture in SIREN, We model the SDF s using an MLP with 4 hidden layers of feature size 256, map a 3D coordinate $\mathbf{p} = (x, y, z)$ to a SDF value: $\mathcal{F}_\theta(\mathbf{p}) = s$. Fourier Feature Networks[26] transform the effective neural tangent kernel (NTK) into a stationary kernel with a tunable bandwidth applying Bochner's theorem. We use a Fourier feature mapping to $\gamma(\mathbf{p}) = [\cos(2\pi \mathbf{B}\mathbf{p}), \sin(2\pi \mathbf{B}\mathbf{p})]^{\mathrm{T}}$, where each entry in $\mathbf{B} \in \mathbb{R}^{m \times d}$ is sampled from $\mathcal{N}\left(0, \sigma^2\right)$, and σ is chosen for each task and dataset with a hyperparameter sweep. In the absence of any strong prior on the frequency spectrum of the signal, we use an isotropic Gaussian distribution:

$$\begin{aligned}\gamma(\mathbf{p}) = [&a_1 \cos\left(2\pi \mathbf{b}_1^{\mathrm{T}} \mathbf{p}\right), a_1 \sin\left(2\pi \mathbf{b}_1^{\mathrm{T}} \mathbf{p}\right), \ldots, \\ &a_m \cos\left(2\pi \mathbf{b}_m^{\mathrm{T}} \mathbf{p}\right), a_m \sin\left(2\pi \mathbf{b}_m^{\mathrm{T}} \mathbf{p}\right)]^{\mathrm{T}}\end{aligned} \quad (7)$$

In addition, we add a network with the same structure in parallel to output the semantic value of each point.

B. Sampling

As mentioned in section III, the M constraints can be cast in loss functions to penalize deviations, where the off-surface point constraints are represented by $\psi(\mathcal{S}(x))$[23], and the corresponding loss function is depicted in Equation (8). Here, $\psi(x) = \exp(-\alpha \cdot |\mathcal{S}(x)|)$, $\alpha \gg 1$ penalizes off-surface points for creating SDF values close to 0. In practice, when new data streams are added, we randomly sample D^t on the whole domain Ω including on-surface points Ω_0 whose SDF values are 0 and off-surface points $\Omega \backslash \Omega_0$ whose SDF values are set to -1.

$$\mathcal{L}_{off} = \int_{\Omega \backslash \Omega_0} \psi(\mathcal{S}(x)) dx \quad (8)$$

On-surface sampling: To make a trade-off between point cloud density and computing efficiency, we choose one key frame from every three frames and feed it to the network for learning and updating the neural network. At each iteration, we sample 75% points from the previous and the rest in the latest key frames to mitigate catastrophic forgetting of past scene. As the scene increases, the proportion of small instances is relatively reduced due to the fixed number of samples, together with the overly redundant samples of buildings and roads, resulting in poor surface fitting of small instances. In view of this, we process importance sampling according to the semantic information to ensure the sampling scale of relevant instances. In every iteration, we sample N_g on-surface points within n_o. In this paper, N_g is 160000, and n_o is 9000 for instances points.

Three-layer Off-surface Sampling: Since the SDF values are set to -1 for all points off the surface, the penalty is the same for points near the surface as for points far away from the surface. However, points in free space are the majority when uniformly sampling, which leads to forgetting the information of the surface. Therefore, the off-surface points will be obtained by three-layer sampling strategy, which is composed of three parts with different proportions:

$$N_g = \lambda_g N_g + \lambda_l N_g + \lambda_n N_g \quad (9)$$

Here, λ_g, λ_l, λ_n are the coefficients corresponding to global sampling, dynamic local sampling and near surface sampling respectively, as is shown in Fig.2.

For a small scene or single object, the scale of the point cloud in all directions remains close, and the off-surface points obtained by global uniform sampling are sufficient to characterize different levels of noise. While in urban scene, with the incremental update of map, the scale difference in

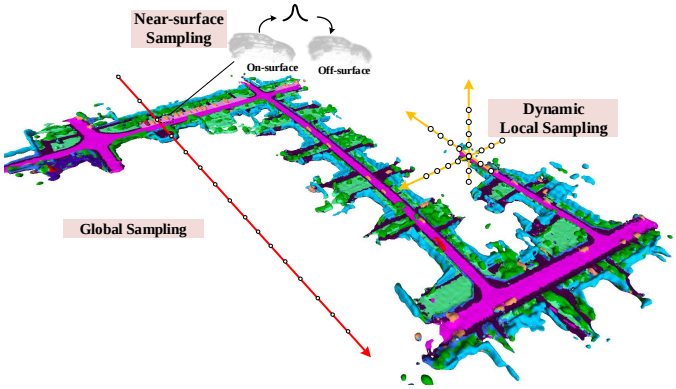


Fig. 2. Three-layer sampling that covers global, dynamic local and near-surface.

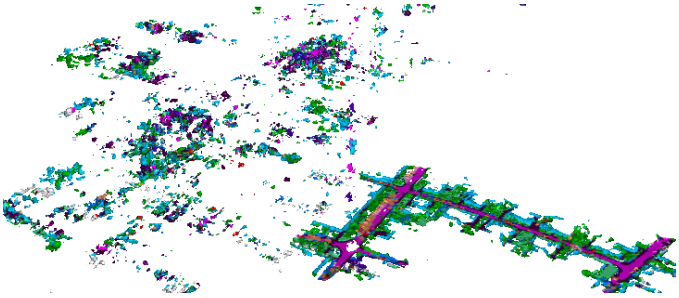


Fig. 3. Only dynamic local sampling will result in the false prediction of SDF value in free space.

Fig. 4. (a) Without near-surface sampling, the network gradually forgets the information of instances in the scene. (b) Near-surface sampling successfully mitigate catastrophic forgetting of instances.

all directions gradually intensifies. In this case, the scale map in $\mathbf{Z}$ direction is almost negligible compared with scale of the scene in $\mathbf{X}$ and $\mathbf{Y}$ direction. Therefore, only the uniform sampling in the whole space is difficult to characterize the noise of local geometry. The results in Fig.8 show that the network with global sampling only learned the overall outline information of the scene lack of detail. And the overall contour information is indispensable, so a certain proportion of points x_g via global sampling is guaranteed:

$$\begin{aligned} \boldsymbol{x_g} &= \mathcal{U}([-1,-1,-1],[1,1,1]), \\ \lambda_g &= len(\boldsymbol{x_g})/len(N_g). \end{aligned} \tag{10}$$

In this paper, λ_g is 0.35.

If global uniform sampling is used throughout the incremental update, most of the off-surface points contribute little at the beginning. Meanwhile, the detail of local geometry is lost. According to the inflow of new LiDAR stream, the dynamic boundary ($\boldsymbol{b_l}$ and $\boldsymbol{b_u}$) of the scene is calculated, where $\boldsymbol{b_l}$ is lower limit of scene boundary and $\boldsymbol{b_u}$ is the upper limit. Then, local off-surface points $\boldsymbol{x_l}$ are sampled uniformly within this range to ensure that the network can learn as much as possible about the local scene changes, $\boldsymbol{x_l} \sim \mathcal{U}(\boldsymbol{b_l}, \boldsymbol{b_u})$.

$$\begin{aligned} \boldsymbol{b_l} &= (\frac{\boldsymbol{L_{min}} - G_{min}}{G_{max} - G_{min}} - 0.5) \times 2 \\ \boldsymbol{b_u} &= (\frac{\boldsymbol{L_{max}} - G_{min}}{G_{max} - G_{min}} - 0.5) \times 2 \end{aligned} \tag{11}$$

Here, $\boldsymbol{L_{max}}(x^l_{max}, y^l_{max}, z^l_{max})$ is the maximum coordinate vector of existing local point clouds, and $\boldsymbol{L_{min}}(x^l_{min}, y^l_{min}, z^l_{min})$ is the minimum one. They change dynamically with the inflow of new data. G_{max} and G_{min} are the maximum and minimum coordinate values of the whole scene. However, local dynamic sampling alone is not sufficient. As off-surface points outside of the local scene do not contribute to the network, which makes the network fail to predict the SDF values in corresponding space, as shown in Fig.3. Hence, a certain proportion of dynamic local sampling points are also required, and λ_l is 0.55 in this paper.

The above two sampling strategies ensure the holistic and dynamic local scene representation, but as the data continues to accumulate, the local sampling will come to be equal to global sampling, except for the spatial scale in the Z direction, causing the network to gradually forget the surface information of the scene, as is shown in Fig.4.

As a result, off-surface points $\boldsymbol{x_n}$ close to the surface of scene $\mathcal{S}$ are sampled to aggravate the penalty of noise near the surface in the learning process:

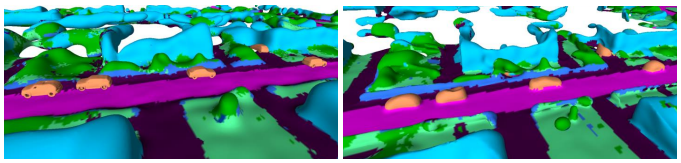
$$\boldsymbol{x_n} = \{(\mathbf{p}+\mathbf{h}, \mathbf{p}-\mathbf{h}) \mid \mathbf{p} \in \mathcal{S}, \mathbf{h} \sim \mathcal{N}(0, \sigma_h)\} \tag{12}$$

where $\mathbf{h}$ is randomly sampled from the multivariate normal distribution $\mathcal{N}(0, \sigma_h) \in \mathbb{R}^3$ with standard deviation σ_h, σ_h is 0.0003 in this paper.

C. Panoptic Representation

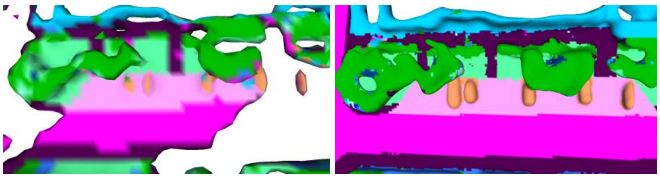
In the scene reconstruction mentioned above, the results of the instances lack finer details. For example, cars look like convex parts on the map, shown in Fig.5(b). One of the core benefits of an object aware approach, is the ability to incorporate inductive bias that objects instances within same category, often have similar 3D shape and appearance. In order to well depict the details of instances in the map, we incorporate category-specific priors by sharing MLP weights across object instances, combined with instance-specific latent codes. Taking the car as an example, we complete instance level reconstruction with finer details in the large-scale scene.

We first use SECOND[31] to detect the bounding box of cars, and transfer the corresponding point cloud from the LiDAR coordinate system to the central coordinate system of the bounding box, $P_B = {T_{LB}}^{-1} \cdot P_L$, where T_{LB} is posture transformation of the bounding box center with respect to the LiDAR coordinate system, P_L is the raw LiDAR data, P_B is the coordinate relative to bounding box center coordinate system. Then we extract the vehicle point cloud P^c from P_B according to the range of bounding box. Next, normalize



(a) (b)

Fig. 5. (a) Hybrid representation of scene and instance. (b) Single scene Representation.



(a) (b)

Fig. 6. (a) Result of traditional inference method. (b) Our method is capable of outputting more details in the same memory.

P^c to [-0.5,0.5] with the boundary of bounding box as the maximum and minimum value and align with the data in ShapeNet. In addition, given fixing θ trained in cars of ShapeNet by DeepSDF[19], a latent code $\hat{\boldsymbol{z}}_i$ for vehicle P_i^c can be estimated via MAP estimation. We concatenate the $\hat{\boldsymbol{z}}_i$ with relative coordinates of vehicle, and then feed it into the MLP of DeepSDF. Adam optimizer is used to update the latent code, and complete the reconstruction of the vehicle instance via Marching Cube according to the SDF value predicted by the network. Finally, the vehicle is scaled and converted to the implicit map coordinate system according to the pose of the center relative to bounding box, the hybrid scene is represented in Fig.5(a).

D. Training and Inference

Training: Besides the constraint $\mathcal{L}_{off}$ described in section IV(B), we construct the rest of the constraints based on the other properties of the SDF[18]: Firstly, $\mathcal{S}(x)$ is differentiable almost everywhere, $-\nabla_x\mathcal{S}$ points towards the the boundary of surface where the gradient exists. If points are sufficiently close to the surface, the gradient $\nabla_x\mathcal{S}$ is equal to the surface normal: $\nabla_x\mathcal{S}(x) = \mathbf{n}(x)$. Secondly, the gradient vector satisfies the Eikonal equation in the whole physical space of interest:$|\nabla_x\mathcal{S}(x)| = 1$. According to the constraints, we randomly sample N_g points from Ω_0, and the same number of off-surface points are sampled in $\Omega\backslash\Omega_0$ by our three-layer sampling strategy, optimizing the SDF with loss:

$$\begin{aligned}\mathcal{L}_{\text{sdf}} = &\int_{\Omega} \|\|\boldsymbol{\nabla}_{\mathbf{x}}\mathcal{S}(x)| - 1\| \, dx \\ &+ \int_{\Omega_0} \|\mathcal{S}(x)\| + (1 - \langle \boldsymbol{\nabla}_x\mathcal{S}(x), \mathbf{n}(x)\rangle) \, dx \\ &+ \int_{\Omega\backslash\Omega_0} \psi(\mathcal{S}(x))dx\end{aligned} \tag{13}$$

In addition, We add a parallel implicit generative head to directly model the implicit semantic label field. Its structure is similar to our SDF model, except that it outputs the probabilities of label classification. We supervise the semantic segmentation and completion results with a multi-classification cross entropy loss:

$$\mathcal{L}_{\text{seg}} = -\frac{1}{N_{\text{seg}}} \sum_{i=1}^{N_{\text{seg}}} \sum_{c=1}^{C} y_{i,c} \log(p_{i,c}). \tag{14}$$

where $y_{i,c}$ and $p_{i,c}$ are the actual and predicted probability for point i belonging to category c respectively. N_{seg} points and C categories are considered.

For panoptic representatio, given a set of cars of ShapeNet, we train a category-shared MLP f_θ follow the DeepSDF by minimizing the sum over losses between the predicted and real SDF values of points in cars under the following loss function:

$$\mathcal{L}\left(f_\theta(\boldsymbol{x}), s\right) = \left|\text{clamp}\left(f_\theta(\boldsymbol{x}), \delta\right) - \text{clamp}(s, \delta)\right|, \tag{15}$$

where $\text{clamp}(x, \delta) := \min(\delta, \max(-\delta, x))$ introduces the parameter δ to control the distance from the surface over which we expect to maintain a metric SDF.

Inference: Comparing with the scales in X and Y direction, scale in Z direction is almost negligible in city map. If we sample the same number of points in all three directions N^3 like the Marching Cube, large resolution will cause the memory explosion, instead, too small resolution means that limited number of points can represent the information on the Z-axis, which will give rise to insufficiency of generated mesh surface to portray the scene details when visualizing. In view of this, we have simply improved the sampling method of Marching Cube. We will sample $N_x \times N_y \times N_z$ points, where $N_x = N_y$, and the N_z is:

$$N_z = \frac{Z_{max} - Z_{min}}{G_{max} - G_{min}} \times N_x. \tag{16}$$

Here, Z_{max} and Z_{min} are the maximum and minimum values of the map on the Z-axis respectively, G_{max} and G_{min} are the maximum and minimum values of the map in all directions. We sample points on the Z-axis from the position Z_{start} where there is information instead of 0, and the Z_{start} is:

$$Z_{start} = \frac{Z_{min} - G_{min}}{G_{max} - G_{min}} \times N_x. \tag{17}$$

Through the above methods, we can ensure the high utilization of memory while generating high-resolution mesh, results in Fig.6 show that our method outperforms the traditional method.

For panoptic representation, given a normalised car detected from LiDAR sequence, we firstly feed the concatenated vector including coordinates and initialized latent code to the f_θ, and optimize the latent code through backpropagation. The coordinates and corresponding latent code with fixed parameters are cascaded into the trained neural network, the SDF values are output and the Marching Cube is used to generate the mesh.

V. Experiments and Results

We train our model on city-scale sequences of SemanticKitti[3] odometry , and the incremental implicit

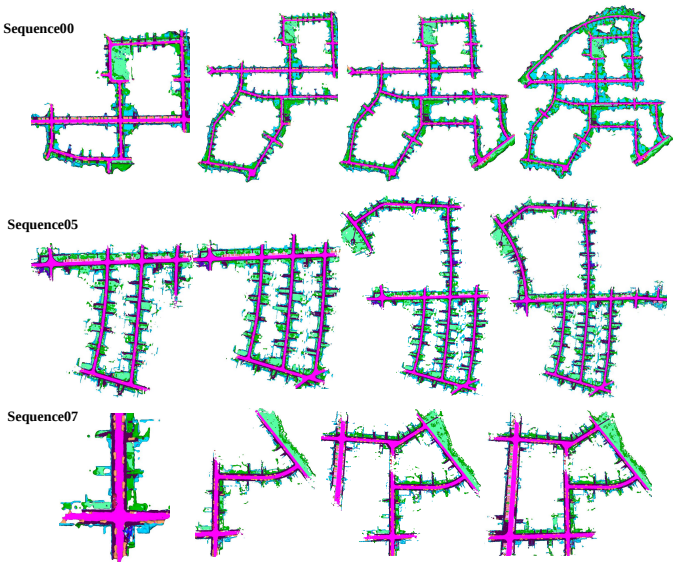


Fig. 7. Incremental implicit semantic mapping of sequence 00, 05 and 07 of SemanticKitti.

mapping is completed on sequence 00, 05 and 07, as is shown in Fig.(7). What's more, the effects of sampling strategy and the encoding methods are further analyzed in the ablation study.

A. Data and Metrics

Data Preparation: We evaluate our method on sequence 00 (4541 scans), 05 (2761 scans) and 07 (1101 scans) of SemanticKitti[3]. Before training, the outliers and dynamic information are deleted in line with the ground truth of semantic label. In addition, to obtain the prior for our cars, we trained the category-specific shared MLP on cars of ShapeNet[7] which consists of 3D points and their SDF values. The reconstructed cars of every frame are extracted according to the range of bounding box detected by the SECOND[31], and the bounding box center is taken as the origin to normalize the point cloud to (-0.5, 0.5), as described in section IV(C). There is a slight deviation in the angle of data obtained by the 3D detection network. Consequently, we fine tune the angle to align our cars as closely as possible with the car in the Shapnet.

Metrics: We evaluate both the depth prediction and semantic segmentation of the system. Despite the absence in geometry detail about road and buildings, it doesn't make much sense to evaluate the completion for them, so we propose a Class-aware Chamfer Distance (CCD). Firstly, we divide the ground truth into 3 categories, one category indicates middle-sized instances including cars and trucks, another includes the rest of the instances of small size, and large scenes like streets and buildings are belong to one category. Then we assign the weight according to the reciprocal of their respective proportions. Finally, we compute mean of chamfer distance of all three classes from the GT point cloud to the predicted mesh respectively, and perform weighting calculation according to the proportion obtained in the previous step. Besides, the averaged interactions over union (mIoU) for semantic segmentation evaluation.

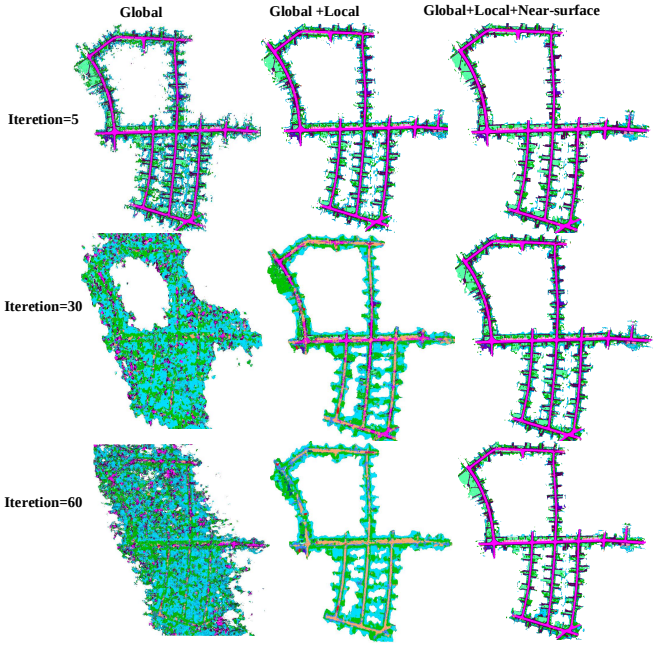


Fig. 8. Ablation study for sampling strategy with different number of iteration

TABLE I
ABLATION STUDY FOR SAMPLING STRATEGY AND ITERATION.

Iteration (epochs)	Global Sampling	Local Sampling	Near-surface Sampling	CCD	mIoU (%)
5	✓	×	×	0.682	67.6
	✓	✓	×	0.672	95.0
	✓	✓	✓	**0.665**	**96.6**
30	✓	×	×	**0.667**	72.4
	✓	✓	×	0.678	74.9
	✓	✓	✓	0.689	**95.6**
60	✓	×	×	0.681	66.3
	✓	✓	×	0.678	66.0
	✓	✓	✓	**0.678**	**96.3**

B. Ablation study for scene representation

Sampling strategy and number of iteration play important roles in implicit reconstruction, prior works mostly focused on the surface reconstruction of small scenes or single objects whose distribution of point clouds in all directions is relatively uniform, and global uniform sampling works in small scenes instead of large scenes. The effectiveness of sampling strategy is analyzed through the ablation study of three layer: global, local, near-surface. The reconstruction results in Fig.(8) show that: Global uniform sampling can easily lead to the over completion of the implicit field in the whole space, which affects the quality of map reconstruction; On this basis, with local sampling, both depth prediction and semantic segmentation will be improved when the number of iteration is small, but when the number of iteration increases, the network will forget the details of the scene; After using the three-layer sampling method, the reconstruction result

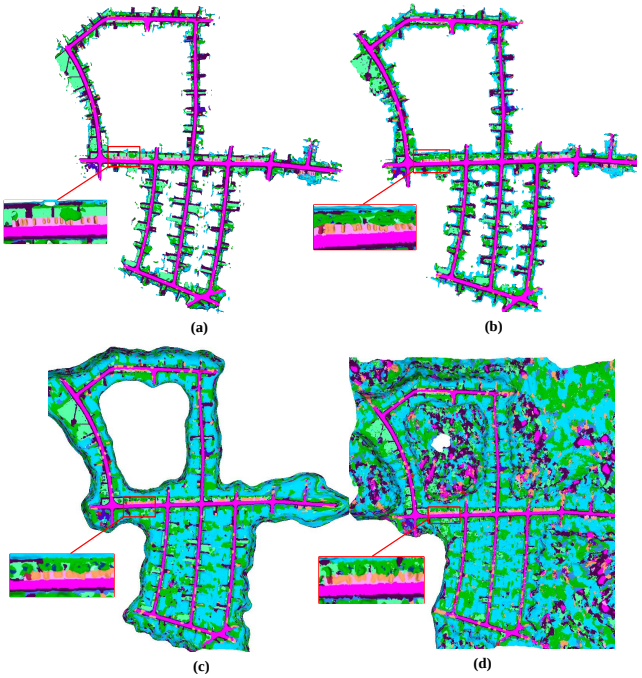


Fig. 9. Representations with (a) Fourier encoding. (b) Positional encoding. (c) Learnable Fourier encoding. (d) no encoding

TABLE II
EVALUATION ON DIFFERENT ENCODING METHODS.

Encoding Method	CCD	mIoU (%)
Fourier	0.665	**96.6**
Positional	0.678	77.8
L-Fourier	**0.532**	90.0
No encoding	0.619	76.0

can remain stable and not affected by other factors. Quantitative evaluation results in TABLE I prove the effectiveness and necessity of this sampling method. Our method perform the best in CCD and mIoU when iteration is 5 and 60, and the evaluation on mIoU is the best when iteration is 30.

Encoding methods: For coordinate-based MLPs, passing input points through a encoding method on a regression task is a prevailing practice. We compare the performance of our task with no input encoding and three encoding methods. One is Positional encoding that is consistent with the work proposed by Rahaman[21] and its encoding level is 10, the other is Fourier encoding[26] with an isotropic Gaussian distribution used in this paper, and the last is Learnable Fourier(L-Fourier) encoding[14] that is the state-of-art method, as is shown in Fig.9. In the case of using the three-layer sampling method, Fourier encoding performs best in semantic segmentation and local details. However, due to the excessive completion of L-Fourier around the ground truth, the point cloud can always find a closer point in reconstructed mesh, and performs best in the quantitative evaluation of depth completion. Due to the wide gap, the CCD can not make up for this deficiency, as is shown in TABLE II.

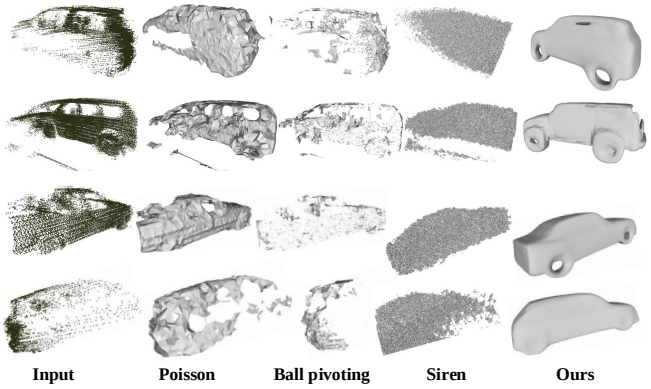


Fig. 10. Qualitative comparison of our method to other methods in Kitti odometry

TABLE III
MEMORY OF DIFFERENT METHODS

Sequence	LOAM	FLOAM	Ours
00	97.5M	313.4M	**1.8**M
05	57.8M	259.3M	**1.8**M
07	20.9M	85.2M	**1.8**M

C. Panoptic Representation

We present a comparison of our Panoptic Representation against two direct (non-learned) methods, Poisson meshing[12] and Ball Pivoting[5], and a learning-based method SIREN[23]. As expected, the direct methods fail to fit a proper shape let alone predict the missing part. Fig. 10. While Poisson meshing barely contains any detail, Ball Pivoting however produces a detailed mesh around the input point cloud but it fails to reconstruct the hidden parts. For SIREN, once there are pieces of missing data, it is easy to cause underfitting and fail to reconstruct the a complete instance. Inspired by the prior of category-specific MLP, our method produce a realistic car shape and reconstruct hidden parts of the cars, which outperforms the other methods.

D. Memory of Map

In order to prove the advantages of implicit maps in memory efficiency, we compared several maps constructed by LiDAR SLAM including LOAM[35] and F-LOAM[28]. As shown in Table III, our method has obvious advantages in terms of efficient memory, and we can render explicit maps with arbitrary resolution as needed to depict more detail of scene.

VI. CONCLUSION

We have presented a city-scale continue learning system with hybrid representation. For scene, when new LiDAR sreams come, three-layer sampling is adopted to ensure the global, local and approximate to surface information learning. For panoptic representation, a category-shared MLP is pre-trained for implicit reconstruction of instances, and achieved the best results in comparision with traditional and learned-based methods. However, the system also suffers

many limitations, now it's not real-time, and there is no state prediction and optimization. In the future, we will concentrate on the SDF-based online pose estimation and optimization for implicit LiDAR SLAM.

REFERENCES

[1] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022.

[2] 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*, pages 6290–6301, 2022.

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019.

[4] Jens Behley and Cyrill Stachniss. Efficient surfel-based slam using 3d laser range data in urban environments. In *Robotics: Science and Systems*, volume 2018, page 59, 2018.

[5] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359, 1999.

[6] Alexandre Boulch, Pierre-Alain Langlois, Gilles Puy, and Renaud Marlet. Needrop: Self-supervised shape representation from sparse point clouds using needle dropping. In *2021 International Conference on 3D Vision (3DV)*, pages 940–950. IEEE, 2021.

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[8] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.

[9] David Hoeller, Nikita Rudin, Christopher Choy, Animashree Anandkumar, and Marco Hutter. Neural scene representation for locomotion on structured terrain. *IEEE Robotics and Automation Letters*, 2022.

[10] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34(3):189–206, 2013.

[11] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020.

[12] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[13] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022.

[14] Yang Li, Si Si, Gang Li, Cho-Jui Hsieh, and Samy Bengio. Learnable fourier features for multi-dimensional spatial positional encoding. *Advances in Neural Information Processing Systems*, 34:15816–15829, 2021.

[15] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[16] 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 *European conference on computer vision*, pages 405–421. Springer, 2020.

[17] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

[18] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. isdf: Real-time neural signed distance fields for robot perception. *arXiv preprint arXiv:2204.02296*, 2022.

[19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[20] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020.

[21] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[22] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12932–12942, 2022.

[23] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

[24] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

[25] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6229–6238, 2021.

[26] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[27] 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.

[28] Han Wang, Chen Wang, Chun-Lin Chen, and Lihua Xie. F-loam: Fast lidar odometry and mapping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4390–4396. IEEE, 2021.

[29] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning shape priors for single-view 3d completion and reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 646–662, 2018.

[30] Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In *International Conference on Neural Information Processing*, pages 264–274. Springer, 2019.

[31] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[32] Zike Yan, Yuxin Tian, Xuesong Shi, Ping Guo, Peng Wang, and Hongbin Zha. Continual neural mapping: Learning an implicit scene representation from sequential observations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15782–15792, 2021.

[33] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021.

[34] Kangxue Yin, Hui Huang, Daniel Cohen-Or, and Hao Zhang. P2p-net: Bidirectional point displacement net for shape transform. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018.

[35] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. 2(9):1–9, 2014.

[36] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022.