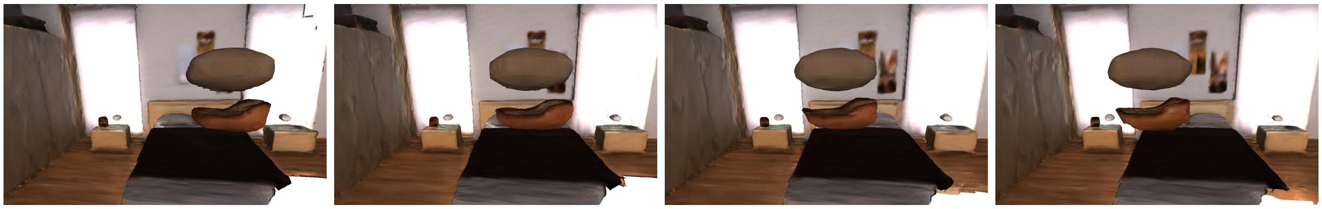
TiV-NeRF: Tracking and Mapping via Time-Varying Representation with Dynamic Neural Radiance Fields

Chengyao Duan[1] and Zhiliu Yang[1]



(a) t = 22.96 s

(b) t = 23.56 s

(c) t = 24.16 s

(d) t = 24.76 s

Fig. 1: Illustration of reconstructing a dynamic object from our NeRF-based dynamic SLAM system. We introduce a dynamic SLAM system capable of camera tracking and moving objects reconstruction. Here, we display the 3D mesh results from Room4 Dataset [1] at different time stamp. It can be clearly seen, the status of flying ship is successfully captured and reconstructed, which is moving from the right side to the left of the room.

Abstract— **Previous attempts to integrate Neural Radiance Fields (NeRF) into Simultaneous Localization and Mapping (SLAM) framework either rely on the assumption of static scenes or treat dynamic objects as outliers. However, most of real-world scenarios is dynamic. In this paper, we propose a time-varying representation to track and reconstruct the dynamic scenes. Our system simultaneously maintains two processes, tracking process and mapping process. For tracking process, the entire input images are uniformly sampled and training of the RGB images are self-supervised. For mapping process, we leverage known masks to differentiate dynamic objects and static backgrounds, and we apply distinct sampling strategies for two types of areas. The parameters optimization for both processes are made up by two stages, the first stage associates time with 3D positions to convert the deformation field to the canonical field. And the second associates time with 3D positions in canonical field to obtain colors and Signed Distance Function (SDF). Besides, We propose a novel keyframe selection strategy based on the overlapping rate. We evaluate our approach on two publicly available synthetic datasets and validate that our method is more effective compared to current state-of-the-art dynamic mapping methods.**

I. INTRODUCTION

Reconstructing an accurate dense map is crucial for tasks such as auto-cars navigation and mixed reality. Existing Simultaneous Localization and Mapping (SLAM) frameworks leverage segmentation neural networks to separate dynamic objects from the scenes, reconstruct dynamic objects and static background in different models. In addition, it always requires a pre-trained model from specific dataset to infer, which limits the application to new environment. Improving the performance of dynamic scenes reconstruction is significant.

Neural Radiance Fields (NeRF) [2] recently attracts a lot of research interest. It can obtain more accurate color and density information by leveraging Multilayer Perceptrons (MLPs). Recently, there are a massive number of works which adapt NeRF to SLAM domain, e.g. iMAP [3] applies a neural implicit representation to traditional dense reconstruction. NICE-SLAM [4] adopts a hierarchical and grid-based neural implicit encoding. However, NICE-SLAM requires to define the size of reconstruction scenes in advance and to provide a pre-trained model. Vox-Fusion [5] exploits octree-based representation and achieves scalable implicit scene reconstruction. The aforementioned methods can reconstruct high-quality map, however, all those works assume the scenarios are static or deem dynamic objects as outliers. Thus, all those works are not able to reconstruct the dynamic objects.

To address this limitation, we propose a novel time-varying representation to track camera poses and reconstruct moving objects in the dynamic scenes, which is named as (TiV-NeRF). The input of our system are RGB-D image sequences with timestamp and segmentation masks of dynamic objects. Inspired by D-NeRF [6], our work extends 3D positions of objects to 4D positions. Then, we design a canonical field, and the other points on the dynamic objects are transformed from deformation field to this canonical field. Next, colors and Signed Distance Function (SDF) of dynamic objects are regressed by a MLP. We maintain two processes simultaneously including tracking process and mapping process, and the two processes are executed in turn. To summarize, the contributions of our paper are as follows:

- We propose, to the best of our knowledge, the first 4D mapping framework to reconstruct the dynamic objects via Neural Radiance Fields.
- We introduce time-varying representation to capture the position offset, which enables our framework to eliminate holes and ghost trail effects of traditional dynamic mapping frameworks.
- We introduce a novel overlap-based keyframe selection strategy to reconstruct more complete dynamic objects.

The rest of our paper is organized as follows: Overview

[1] C. Duan and Z. Yang are with the School of Information Science and Engineering, Yunnan University, Kunming, Yunnan, China

of related work is discussed in Section II. We provide a detailed explanation of our method in Section III. In Section IV, we demonstrate our experiment results including camera tracking, reconstruction quality, object completion ability, and ablation study. In the end, we summarize the experimental results in Section V.

II. RELATED WORK

Dense SLAM & Dynamic SLAM. Classic dense SLAM systems have developed rapidly in the past decades. KinectFusion [7] introduces a dense reconstruction and tracking system based on Truncated Signed Distance Function (TSDF) with a RGB-D camera. ElasticFusion [8] is a surface-based SLAM system, and proposes two states, defined as active and inactive, to control the activation state of voxel.

However, aforementioned systems solely focus on static scenes. Complete static environments are rare in real-world scenarios. To this end, Co-Fusion [1] maintains a background model and multiple dynamic foregrounds, detecting moving objects through motion filtering and semantic segmentation. MaskFusion [9] utilizes Mask R-CNN [10] to achieve more precise segmentation of dynamic objects. MID-Fusion [11] introduces the use of Volume TSDF for dense mapping and tracking, but it involves at least four raycastings, leading to slow processing. EM-Fusion [12] proposes a tracking method based on a probabilistic expectation maximization (EM) formulation for reconstructing dynamic objects. TSDF++ [13] advocates using a single large reconstruction volume to store the entire dynamic scene.

RF [14] leverages motion priors and treats all dynamic objects as one rigid body. ACEFusion [15] adopts a hybrid representation including octrees and surfels. The above-mentioned methods can track and reconstruct for dynamic scenes, but those systems come with certain limitations, such as, leading to ghost trail effect and requiring pre-trained models.

Dynamic NeRFs. NeRF is firstly proposed for static scenes. A large number of recent studies extend its application to dynamic environments. D-NeRF [6] is a widely adopted solution which associates time with 3D position and establishes a transformation between deformation field and canonical field. NeRF-W [16], Nerfies [17], HyperNeRF [18] associate a latent deformation code and an appearance code rather than time to each image, realizing the transformation same to D-NeRF. NR-NeRF [19] employs MLPs to disentangle dynamic scene into a canonical volume and deformation volume. Additionally, it employs a rigid network to constrain rigid regions. NSFF [20] is proposed for handling varieties of in-the-wild scenes, including thin structures, view-dependent effects and complex degrees of motion. D^{2}NeRF [21] introduces a novel direction to decouple dynamic and static objects from a monocular video. NeRFPlayer [22] proposes a decomposition of dynamic scenes based on their temporal characteristics. [23] utilizes an additional time parameter and executes temporal feature interpolation. RoDynRF [24] proposes a space-time synthesis algorithm from a dynamic monocular video and acquires accurate camera pose for dynamic environments with high-speed moving objects. A. Yu et al. [25] presents a novel PlenOctrees to achieve real-time reconstruction and L. Wang et al. [26] extends it to dynamic scenes.

NeRFs-based SLAM. The power of NeRF in synthesizing photo-realistic novel views relies on accurate camera poses. Thus, there are some attempts to integrate NeRF into SLAM. iNeRF is the first work to verify the fact that it is possible to obtain camera poses from a carefully trained NeRF field. Barf [27] further improves iNeRF, by proposing a method to simultaneously train a NeRF and estimate the camera pose.

iMAP [3] is the first dense real-time SLAM system that leverages an implicit neural scene representation and is able to solve the problem of camera localization. DROID-SLAM [28] can be trained only with monocular input and then directly applied to stereo or RGB-D input, it can obtain improved accuracy for stereo or RGB-D cases without retraining. NICE-SLAM [4] adopts a local-update strategy and proposes a hierarchical and grid-based neural implicit encoding, but it requires a pre-trained model. Vox-Fusion [5] proposes an accurate and effective SLAM system based on sparse voxel octree. It combines voxel embedding to alleviate artifacts of voxel borders. NICER-SLAM ??? is the RGB-image-only dense SLAM system and it proposes a local adaptive transformation for signed distance functions (SDFs). Co-SLAM [29] designs a joint coordinate and sparse grid encoding for input points. GO-SLAM [30] proposes a real-time global pose optimization system that considers the complete history information of input frames and incrementally aligns all poses.

However, there is a critical drawback for those existing NeRF-based SLAMs, that is all of these systems are based on the assumption of static scenes. Our framework aims to address this issue and construct the dynmic objects in the real-time.

III. METHODOLOGY

Given a stream of continuous RGB-D frames with color images, depths and corresponding masks, we simultaneously maintain two processes, tracking process and mapping process, in our framework. The overview of our TiV-NeRF framework is shown in Fig. 2.

A. Neural Deformation Fields

We define the status under the $t = 0$ as the canonical field and the status of other time as deformation field. Inspired by D-NeRF [6], we leverage a deformation network Θ_d to map all scene positions from a deformation field to a canonical field. Specifically, we utilize Θ_d to regress offsets which represents the embeddings of every point for every frame $i \in \{1, ..., n\}$, where n is the number of input images and offset is zero only if $t = 0$. $e(t)$ is trilinearly interpolating result of voxel embeddings. The metric can be expressed as follows:

$$\Theta_d(\mathbf{e}(t), t) = \begin{cases} \Delta e & \text{if } t \neq 0, \\ 0 & \text{if } t = 0. \end{cases} \tag{1}$$

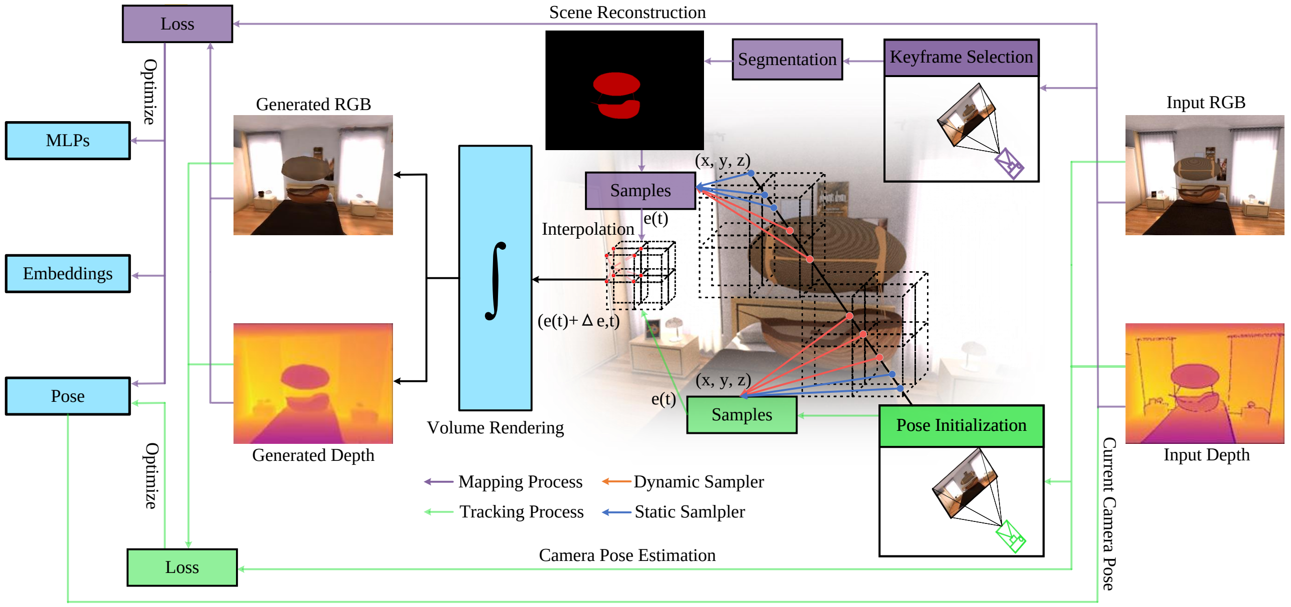


Fig. 2: Overview of our TiV-NeRF framework. Our system simultaneously maintains two processes, tracking process and mapping process. **Tracking Process:** It will first estimate the camera pose of the current frame and then samples valid points from the scenes. Subsequently, it will encode these points by a set of N-dimensional embedding vectors and interpolation is achieved correspondingly. The interpolated results are used as input for the MLPs. Finally the colors and SDF are predicted by MLPs, and the RGB images and depth images are rendered to compute tracking loss and optimize the pose of current frame. **Mapping Process:** Firstly, after acquiring the camera pose of current frame, we design a strategy to select target keyframes for reconstruction. Keyframes are selected from a incrementally-built database. Then a bunch of points are sampled from the scenes. Then aforementioned procedures, including embedding, interpolation, MLPs regression, are executed to obtain colors and SDF values, which will be used to reconstructing map. At the same time, the poses, embedding parameter and MLPs are optimized.

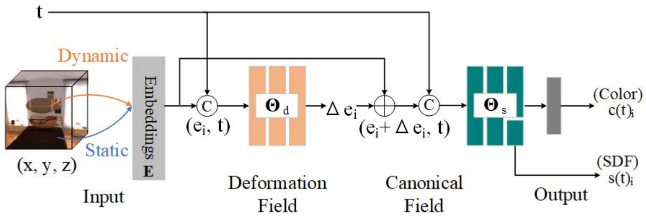


Fig. 3: We simultaneously sample points in static background and dynamic foregrounds, maintaining a collection of N-dimensional sparse voxel embeddings like Vox-Fusion and encoding positions **x** = (x, y, z) using this collection to obtain e(t). Then we will associate e(t) with time t_i to regress offsets Δe by Θ_d. In the end, we obtain colors and SDF by Θ_s.

Accordingly we employ a MLP to obtain colors and SDF. The specific content of this part is shown in Fig. 3. Notably, due to SDF values instead of volume densities that we acquire, the prime volume rendering formula in NeRF becomes inapplicable. Thus we adopt the volume rendering formula that proposed in [31] and modified in Vox-Fusion [5].

However, in order to represent dynamic scenes, we have modified the formula as follows:

$$\mathbf{e}(t) = \mathrm{TriLerp}(\hat{u}_i T_i \mathbf{x}(t)_i, \mathbf{E}). \tag{2}$$

$$\mathbf{c}(t)_i, s(t)_i = \Theta_s((\Theta_d(\mathbf{e}(t), t) + \mathbf{e}(t)), t). \tag{3}$$

where **E** is the N-dimensional collection and TriLerp$(\cdot,\cdot)$ denotes the trilinear interpolation function which yields interpolated embeddings $\mathbf{e}(\mathbf{t})$. $\hat{u}_i$ signifies that the pose of current frame is updated based on the previous one. Additionally, $\mathbf{c(t)_i}$ and $s(t)_i$ are the colors and SDF of these points along each camera ray respectively.

$$\omega(t)_i = \sigma(\frac{s(t)_i}{tr}) \cdot \sigma(-\frac{s(t)_i}{tr}). \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid activation function and $\omega(t)_i$ is a weight parameter that when $\sigma(\frac{s(t)_i}{tr}) = \sigma(-\frac{s(t)_i}{tr})$, $\omega(t)_i$ get a maximum value. It means that there is a larger weight around object surface.

$$\mathbf{C}(t) = \frac{1}{\sum_{i=0}^{N-1} \omega(t)_i} \sum_{i=0}^{N-1} \omega(t)_i \cdot \mathbf{c}(t)_i. \tag{5}$$

$$D(t) = \frac{1}{\sum_{j=0}^{N-1} \omega(t)_i} \sum_{i=0}^{N-1} \omega(t)_i \cdot d(t)_i. \tag{6}$$

Finally, $\mathbf{c(t)_i}$ and $d(t)_i$ express the colors and SDF of i-th point along ray, thus the the color and depth are calculated through weighted summation. As shown in $\mathbf{C}(t)$ and $D(t)$.

B. Loss Function

We divide all sampling pixels into different batch b and there are B batches in total. we sample N pixels at every batch which every pixel will generate a ray. We utilize loss function just like [31] and Vox-Fusion [5] as follows:

$$\begin{aligned}\mathcal{L}^b(t) = \sum_{b=0}^{B} (\lambda_{color}\mathcal{L}^b_{color}(t) + \lambda_{depth}\mathcal{L}^b_{depth}(t) \\ + \lambda_{space}\mathcal{L}^b_{space}(t) + \lambda_{SDF}\mathcal{L}^b_{SDF}(t) + \lambda_{zero}\mathcal{L}^b_{zero}(t).\end{aligned} \tag{7}$$

where the $\mathcal{L}^b(t)$ consist of five parts: color loss, depth loss, free-space loss, SDF loss and offset loss. These five parts of loss function will be multiplied by their respective coefficients, and finally added together. The $\mathcal{L}^b_{zero}(t)$ is designed to minimize the potential offset of the background as much as possible because we believe that the position of background should remain fixed.

$$\mathcal{L}^b_{color}(t) = \frac{1}{N}\sum_{i=0}^{N-1}\|\mathbf{C}_i(t) - \mathbf{C}_i^{gt}\|. \tag{8}$$

$$\mathcal{L}^b_{depth}(t) = \frac{1}{N}\sum_{i=0}^{N-1}\|D_i(t) - D_i^{gt}\|. \tag{9}$$

where $\mathcal{L}^b_{color}(t)$ and $\mathcal{L}^b_{depth}(t)$ are the loss function of color and depth of sampling points, we predict color and depth at time t using Θ_s, then we can compute loss values between predicted values and the ground-truth values.

$$\begin{aligned} \mathcal{L}^b_{space}(t) &= \beta_{space}\sum_{s\in S_p^{space}}(D_s(t) - tr)^2, \\ where \quad \beta_{space} &= (1 - \frac{N_p^{space}(t)}{N_p^{space}(t) + N_p^{tr}(t)}). \end{aligned} \tag{10}$$

$$\begin{aligned} \mathcal{L}^b_{SDF}(t) &= \beta_{SDF}\sum_{s\in S_p^{tr}}(D_s(t) - D_s^{gt})^2, \\ where \quad \beta_{SDF} &= (1 - \frac{N_p^{tr}(t)}{N_p^{space}(t) + N_p^{tr}(t)}). \end{aligned} \tag{11}$$

where $N_p^{tr}(t)$ and $N_p^{space}(t)$ are the number of points in free space and near the surface of the object respectively. $s \in S_p^{space}$ express those points sampled in the free space (i.e. in front of the object surface) and $s \in S_p^{tr}$ is the point which is closed to object surface. Specifically, D_s, the depth of $s \in S_p^{space}$, will learn towards truncation tr in $\mathcal{L}^b_{space}(t)$ which tr represents positions away from surface in TSDF values. In $\mathcal{L}^b_{SDF}(t)$, D_s, the depth of $s \in S_p^{tr}$, is constrained within the truncation range of the object surface.

C. Robust Camera Pose Estimation

On the tracking process, we fix the embeddings model parameters and MLPs (Θ_d and Θ_s), that is to say that we just optimize the pose $T_i \in SE(3)$ of current frame. Specifically, we sample rays in the whole input image, and then acquiring the predicted colors and SDF. In the end, we compute the loss described in Equation (7) between predicted values and ground-truth values. $\hat{u}_i \in SE(3)$ is a relative pose transformation matrix from the last frame to the current frame. In this paper, we adopt a zero-motion model, i.e. we define the pose of last frame as the pose of current frame in tracking process.

We find that tracking pose will become unstable through experiments after associating positions with time t. We try to improve the robustness of tracking process and abandon the loss compared to the last tracking loss. It is described as follows:

$$min\{\mathcal{L}^b_{last}(t), \lambda_r\mathcal{L}^b_{current}(t)\}. \tag{12}$$

where it will abandon the more inaccurate tracking loss comparing to the loss of last tracking.

D. Dynamic Mapping

*1) **Keyframe Selection**:* We maintain a keyframes database and insert keyframe at a fixed gap. Randomly selecting keyframes from the keyframes database as target keyframes to optimize the global map used in Vox-Fusion [5] leads to too many uncertainties and we verify this issue experimentally. Thus we propose a novel keyframe selection strategy. We calculate the overlap ratio between the current frame and each frame in keyframes database by casting rays. Specifically, we convert the sampling points obtained from the current frame into pixel coordinates and then calculate the ratio of these sampling points falling in the each keyframe of keyframes database as the overlap ratio. Regarding the N_k frames with the lowest overlap ratio as the target keyframes gives a competitive result. The purpose of what we do is to let the target keyframes cover as many voxels as possible in the scenes.

*2) **Optimization**:* Different from the tracking process, our mapping process involves the simultaneous optimization of embedding parameters, MLPs, and the poses of target keyframes. We adjuste our approach to optimize parameters based on the loss of each individual keyframe, rather than relying on the loss of the entire group of target keyframes.

IV. Experiments

A. Experimental Setup

*1) **Implementation Details**:* We employ two MLPs in our experiment, denoted as Θ_d, which regresses the offsets of the points, and Θ_s, which regresses the colors and sdf of the points. All experiment results are evaluated on an NVIDIA RTX 4090 GPU. Specifically, Θ_d and Θ_s both consist of three hidden layers, the sdf is obtained from the third hidden layer of Θ_s. Moreover, the tracking process of a single frame is configured to run for 30 iterations and the mapping process for 15 iterations. Initially, the camera pose is set as an identity matrix. The size of input images are (640, 480), we down-sample the image to (320, 240) for our rendering experiments.

*2) **Datasets**:* We evaluate our system on two different public synthetic datasets (i.e. Room4 and ToyCar3) which are provided by Co-Fusion [1]. Both of them contain multiple dynamic objects. These datasets provide RGB-D sequences with ground truth camera trajectory and the masks of dynamic objects. Additionally, these datasets include a Kinect camera-captured video, from which we extract timestamps.

TABLE I: Camera Tracking Results

Methods	ATE	Room4	ToyCar3
NICE-SLAM [4]	RMSE [m][(↓)	0.1305	0.0785
	Mean [m](↓)	0.0680	0.0617
	Std [m](↓)	0.1113	0.0486
Vox-Fusion [5]	RMSE [m](↓)	0.0164	0.0655
	Mean [m](↓)	0.0147	**0.0543**
	Std [m](↓)	0.0073	0.0366
Ours	RMSE [m](↓)	**0.0131**	**0.0653**
	Mean [m](↓)	**0.0118**	0.0548
	Std [m](↓)	**0.0056**	**0.0355**

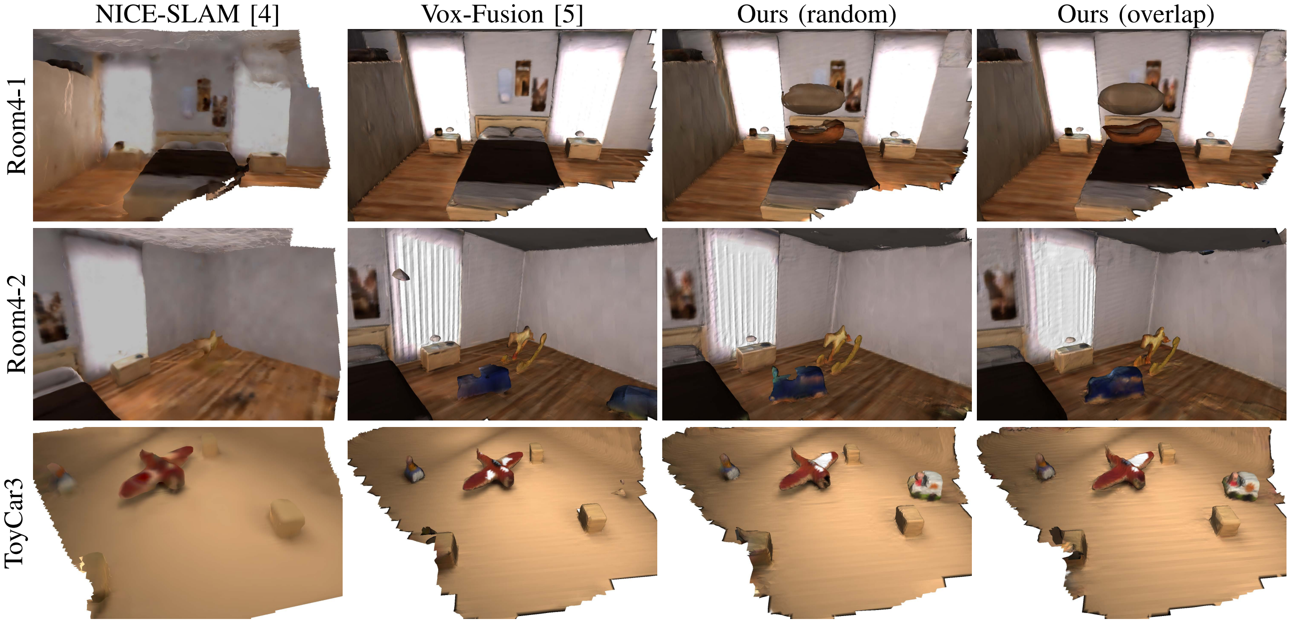


Fig. 4: Comparison of Mapping Quality Results. Room4-1. From the experimental results, it can be observed that we have completely reconstructed the flying ship. NICE-SLAM [4] and Vox-Fusion [5], however, are unable to reconstruct dynamic objects. Randomly selecting the keyframes will be unstable and sometimes there are gaps in the reconstructed objects. Ours based on overlap avoids this problem. **Room4-2.** The results indicate that NICE-SLAM cannot reconstruct the blue car at all, and Vox-Fusion, as confirmed by our validation, can only occasionally reconstruct the dynamic car while most of the time it fails to do so. In addition Vox-Fusion will reconstruct dynamic objects in inappropriate places. When the blue car has moved to the bedside, it reconstructs it in its original position. Similarly, our method will encounter the above problems when using two different keyframe selection strategies, but the overlap-based method will have better results. **ToyCar3.** Obviously, Our system have reconstructed the white car, but others treat it as an outlier and ignore it. Addditional, the white portion on the wing is a result of reflection and it should have already shifted to the left wing completely. Although we do not have the known masks of this reflection, our method still effectively captures this transitional process.

B. Evaluation of Camera Tracking

We calculate the Absolute Trajectory Error (ATE) to evaluate the accuracy of camera tracking. As shown in Table I, we can clearly observe that our system predicts the most accurate camera pose in dynamic scenes compared to NICE-SLAM and Vox-Fusion. Specifically, we successfully reconstruct the dynamic objects in the scenes, which is also beneficial for estimating more precise camera poses. As described in Equation (12), we add a checking stage to enhance the robustness of camera pose estimation, and results validate this enhancement is effective in dynamic scenes. If this checking stage is not applied, the camera pose estimation is unstable.

C. Mapping Quality Evaluation

*1) **Qualitative Analysis**:* We qualitatively compare the reconstruction results on NICE-SLAM [4], Vox-Fusion [5] and ours shown in Fig. 4. Ours and Vox-Fusion do not require pre-trained model, whereas NICE-SLAM does. Obviously, we have better results in the reconstruction of dynamic objects. NICE-SLAM proposed a robust method to reconstruct scenes, but it still treats the dynamic objects as outliers and it will try to avoid reconstructing these dynamic objects as much as possible. Vox-Fusion only occasionally reconstructs a part of inaccurate dynamic objects and there are even some dynamic objects that are not reconstructed at all. We will further analyze the differences between the two different keyframe selection strategies in Section. IV-D.

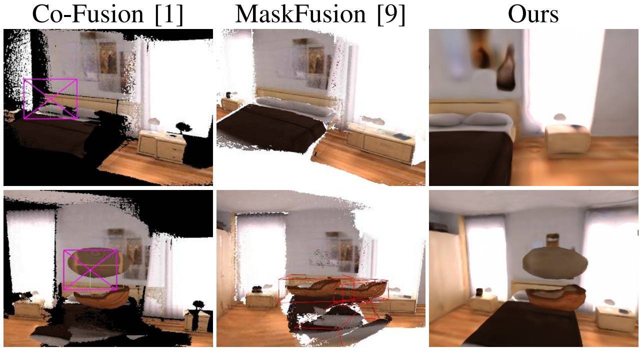


Fig. 5: Existing Issues of Traditional SLAM systems. We demonstrate experimental results obtained using Co-Fusion and MaskFusion, both of which rely on pre-trained models. Throughout the reconstruction process, these methods produce numerous inaccurate reconstructions (e.g., nightstand and ship) and gaps in the reconstructed scenes (black and white regions) compared to our system. This motivates our exploration of applying dynamic NeRF to SLAM.

Traditional SLAM methods have the ability to address the problem of pose estimation, but these methods will bring a series of problems like ghost trail effects and gaps. these issues will have a detrimental impact, especially in the context of large-scale 3D reconstruction. We demonstrate some of these issues through experiments and results can be seen in Fig. 5. Furthermore, comparing our experimental results with Co-Fusion and MaskFusion, our system avoids the issues commonly encountered in traditional SLAM systems.

*2) **Quantitative Analysis**:* We calculate the **MSE** (Mean Squared Error), **PSNR** (Peak Signal-to-Noise Ratio), **SSIM** (Structural Similarity Index) and **LPIPS** (Learned Perceptual

Image Patch Similarity) and the calculated results are shown in the Table. II. We find our method struggles with these indicators on Room4. Through the observation of the visual reconstruction results, we guess it might be because we do not distinguish between the foreground and background, associating all points with time t. This is because that we try to utilize single MLP to represent the entire scene. However, single MLP has limitations in expressing the size of the scene especially we simultaneously reconstruct dynamic and static elements. However we can obtain competitive quantitative results on ToyCar3 compared to others because of the smaller scene.

While we may not achieve the best performance in these qualitative measurements, it is evident from the quantitative results in Fig. 4 that our superiority lies in our capability to completely and precisely reconstruct dynamic objects compared to other methods.

TABLE II: Novel View Synthesis Evaluation

Methods	Metrics	Room4	ToyCar3
NICE-SLAM [4]	MSE(↓)	0.0114	0.0045
	PSNR(↑)	22.1614	24.1507
	SSIM(↑)	0.6755	0.8950
	LPIPS(↓)	0.5983	0.2851
Vox-Fusion [5]	MSE(↓)	**0.0040**	0.0033
	PSNR(↑)	**25.5220**	25.1758
	SSIM(↑)	**0.7882**	**0.9158**
	PLIPS(↓)	**0.3969**	0.1891
Ours (random)	MSE(↓)	0.0045	0.0034
	PSNR(↑)	23.6914	24.6352
	SSIM(↑)	0.7174	0.8926
	LPIPS(↓)	0.4166	0.1881
Ours (overlap)	MSE(↓)	0.0045	**0.0027**
	PSNR(↑)	23.6628	**25.7592**
	SSIM(↑)	0.7139	0.9081
	LPIPS(↓)	0.4286	**0.1778**

D. Object Completion

We find that the keyframe selection strategy used in Vox-Fusion [5] is unstable. By comparing the results of multiple experiments, the results of each experiment will be different. Thus we propose a novel keyframe selection strategy, calculating the overlap between current frame and the keyframes database. Surprisingly, we have achieved outstanding completeness in reconstructing dynamic objects. We have shown the results of the two different keyframe selection strategies for reconstructing the blue car in Fig. 6. It's evident that there are gaps on the hood of the blue car, and it's empty in the direction away from the perspective of camera. However, the ability of novel view synthesis is limited, we cannot completely fill the part we have ever observed.

E. Ablation Study

In this paper, we correlate 3D positions with the time t and increase the number of sampling points within known dynamic masks. To validate the effectiveness of our approach, we conduct a series of ablation experiments, the results of which are depicted in Fig. 7, merely illustrating the outcomes of our ablation study at the 609-th frame. Specifically, we conduct experiments where we individually omit time t and dynamic masks, subsequently comparing their results with our approach. It is evident that our system, which associates positions with time and leverages dynamic masks, is a well-grounded and reasoned strategy.

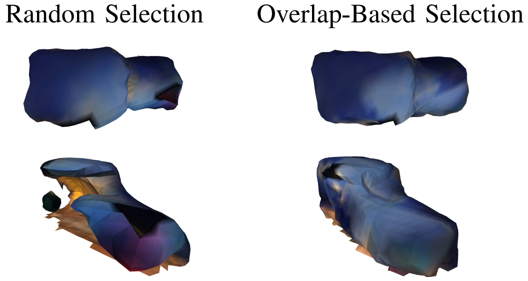


Fig. 6: Object Completion. Keyframe selection strategy used in Vox-Fusion cannot reconstruct parts that do not appear in the current view, ours based on overlap, however, will completely reconstruct the blue car even we have not observed the part of the car.

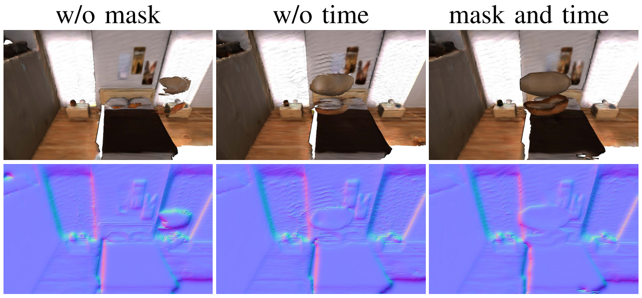


Fig. 7: Ablation Study. Here are the experimental results that do not include the dynamic masks and the time t. Notably we ensure the uniqueness of the variables. When we remove the dynamic masks, it can be observed that the reconstructed dynamic ship is incomplete and appears in incorrect positions. When we remove the time t, the result shows that it can only occasionally produce general results that there are many holes on the ship and distorted lines on the wall.

V. CONCLUSIONS

This paper introduces a novel dense SLAM system that leverages neural implicit representations in the dynamic domain. Our novelty lies in the introduction of a time-varying representation to extend 3D space positions to 4D space-temporal positions. Additionally, we introduce a more efficient keyframe selection strategy, which we calculate the overlap ratio between the current frame and each frame in keyframes database by casting rays. And our keyframe selection strategy based on the overlapping-ratio enables us to construct more complete dynamic objects. Different from existing approaches, our method do not rely on pre-trained models to reconstruct the dynamic objects.

Limits & Future work. Our system presents a competitive performance on multiple public dynamic synthesis datasets, while evaluations on the real-world sequences are absent due to the limited access to public dynamic datasets. Furthermore, our method enrolls inaccurate camera pose estimation when there are high-speed moving objects in the environment. These are potential directions for our next step work.

References

[1] M. Rünz and L. Agapito, "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4471–4478.

[2] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[3] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.

[4] 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.

[5] 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.

[6] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 318–10 327.

[7] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. 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*, 2011, pp. 559–568.

[8] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph." Robotics: Science and Systems, 2015.

[9] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2018, pp. 10–20.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[11] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "Mid-fusion: Octree-based object-level multi-instance dynamic slam," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5231–5237.

[12] M. Strecke and J. Stuckler, "Em-fusion: Dynamic object-level slam with probabilistic data association," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5865–5874.

[13] M. Grinvald, F. Tombari, R. Siegwart, and J. Nieto, "Tsdf++: A multi-object formulation for dynamic object tracking and reconstruction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 192–14 198.

[14] Y.-S. Wong, C. Li, M. Niessner, and N. J. Mitra, "Rigidfusion: Rgb-d scene reconstruction with rigidly-moving objects," in *Computer Graphics Forum*, vol. 40, no. 2. Wiley Online Library, 2021, pp. 511–522.

[15] M. Bujanca, B. Lennox, and M. Luján, "Acefusion-accelerated and energy-efficient semantic 3d reconstruction of dynamic scenes," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 063–11 070.

[16] 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.

[17] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5865–5874.

[18] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields," *arXiv preprint arXiv:2106.13228*, 2021.

[19] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, and C. Theobalt, "Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 959–12 970.

[20] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6498–6508.

[21] T. Wu, F. Zhong, A. Tagliasacchi, F. Cole, and C. Oztireli, "Dˆ 2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 653–32 666, 2022.

[22] L. Song, A. Chen, Z. Li, Z. Chen, L. Chen, J. Yuan, Y. Xu, and A. Geiger, "Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2732–2742, 2023.

[23] S. Park, M. Son, S. Jang, Y. C. Ahn, J.-Y. Kim, and N. Kang, "Temporal interpolation is all you need for dynamic neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4212–4221.

[24] Y.-L. Liu, C. Gao, A. Meuleman, H.-Y. Tseng, A. Saraf, C. Kim, Y.-Y. Chuang, J. Kopf, and J.-B. Huang, "Robust dynamic radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13–23.

[25] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5752–5761.

[26] L. Wang, J. Zhang, X. Liu, F. Zhao, Y. Zhang, Y. Zhang, M. Wu, J. Yu, and L. Xu, "Fourier plenoctrees for dynamic radiance field rendering in real-time," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 524–13 534.

[27] T. Barf and A. Kaptein, "Irreversible protein kinase inhibitors: balancing the benefits and risks," *Journal of medicinal chemistry*, vol. 55, no. 14, pp. 6243–6262, 2012.

[28] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, pp. 16 558–16 569, 2021.

[29] 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.

[30] Y. Zhang, F. Tosi, S. Mattoccia, and M. Poggi, "Go-slam: Global optimization for consistent 3d instant reconstruction," *arXiv preprint arXiv:2309.02436*, 2023.

[31] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural rgb-d surface reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6290–6301.