

# Self-Supervised Multimodal NeRF for Autonomous Driving

Gaurav Sharma<sup>1</sup>, Ravi Kothari<sup>2</sup>, Dr. Josef Schmid<sup>3</sup>

**Abstract**— In this paper, we propose a Neural Radiance Fields (NeRF) based framework, referred to as Novel View Synthesis Framework (NVSF). It jointly learns the implicit neural representation of space and time-varying scene for both LiDAR and Camera. We test this on a real-world autonomous driving scenario containing both static and dynamic scenes. Compared to existing multimodal dynamic NeRFs, our framework is self-supervised, thus eliminating the need for 3D labels. For efficient training and faster convergence, we introduce heuristic based image pixel sampling to focus on pixels with rich information. To preserve the local features of LiDAR points, a Double Gradient based mask is employed. Extensive experiments on the KITTI-360 dataset show that, compared to the baseline models, our Framework has reported best performance on both LiDAR and Camera domain. Code of the model is available at <https://github.com/gaurav00700/Selfsupervised-NVSF>

## I. INTRODUCTION

Light Detection and Ranging (LiDAR) and camera represent two of the most critical sensors in the perception systems of Autonomous Vehicle (AV). However, the collection and annotation of such data is an inherently labor-intensive and costly endeavor, necessitating substantial manual effort and resources. The generation of highly accurate synthetic data offers a promising alternative, potentially circumventing these challenges. There are many model-driven approaches available for generating novel view synthetic LiDAR and Camera data based on scene formulation, such as Game engine based [1], [2], Explicit model based [3], [4] and Implicit model based [5], [6]. All existing methods come with their own set of limitations. For instance, CARLA [1] operates within a handcrafted environment, while some approaches fail to construct view-dependent attributes like raydrop and point cloud intensity or rely on explicit scene models. As a result, these methods struggle to generalise effectively, particularly in the LiDAR domain, due to the challenges in accurately mapping the physics of active sensors. Achieving accurate and realistic Novel View Synthesis (NVS) is also crucial for minimizing the sim-to-real domain gap.

Advancements in differential volume rendering and NeRF based methods have ushered in new possibilities in NVS. Thus, we propose the NVSF framework, leveraging NeRF to synthesise novel views for both LiDAR and camera. Central to the proposed framework is joint learning of implicit neural representation of scene geometry and view-dependent attributes for both LiDAR and Camera sensors. Exploration of LiDAR domain adaptation through the synthesis of Point cloud data (PCD) using NeRF spans both static

The author is with the AVL Software and Functions GmbH, Germany and Technische Hochschule Deggendorf, Germany. <sup>1</sup>gaurav.sharma@avl.com, <sup>2</sup>ravi.kothari@avl.com, <sup>3</sup>josef.schmid@th-deg.de

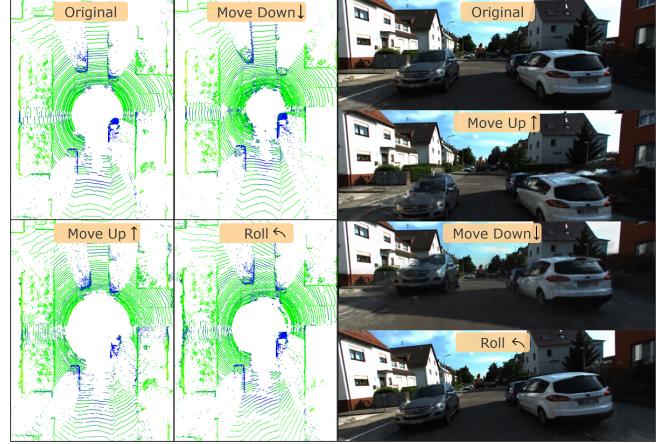


Fig. 1: Joint Novel View Synthesis for Camera and LiDAR sensor by the proposed method. Along with extrinsic attributes, intrinsic sensor parameters can be also varied.

and dynamic scenes, with the latter posing unique challenges due to NeRF's inherent operating principles. To address the dynamic scene problem, the proposed framework draws inspiration from LiDAR4D [7] and K-Planes [8], which factorise a canonical scene, 4D volume of space and time through multiple 2D planes. It decomposes a canonical Scene in space and time, thereby making the model interpretable between static and dynamic objects. Unlike Neural Scene Graph (NSG) [9] based methods, the proposed framework does not require any external supervision from 3D annotation to decompose a canonical scene into corresponding foreground and background scenes. To resolve this K-Planes based network handles dynamic scene of both non-rigid and rigid objects such as pedestrians, cars, trucks, etc. The proposed framework empowers the novel view synthesis of LiDAR and camera in real-world driving scenarios of AVs, supporting diverse applications such as minimizing domain gaps, domain invariant training, experimenting with different sensor specifications, sensor configurations on AV and revival of corrupted data.

Our contributions are threefold:

- 1) We propose a multimodal self-supervised NVS framework which jointly learns the implicit Neural representation of a 4D spatio-temporal scene for LiDAR and Camera
- 2) We introduce a Heuristic pixel sampling method using Multinomial Probability Distribution based on reconstruction error of image pixels during training
- 3) To improve feature alignment in synthesised LiDAR

point clouds, we propose a Double Gradient-based clipping mask for capturing sparse features

## II. RELATED WORK

(a) *NeRF*: NeRF [5] and its derivatives [10]–[12] have demonstrated remarkable performance in 3D view reconstruction using multi-view 2D images. Conventional NeRF relies on Multi Layer Perceptron (MLP) to learn the scene content at each queried point, but the need to compute thousands of points along a ray significantly slows down the learning process. Instant-NGP (iNGP) [13], TensoRF [14] and K-Planes [8] employ learnable embeddings or voxels to represent 3D scenes, which significantly reduce the computational burden of the rendering MLP. Recent research efforts centered on NeRFs have driven significant advancements and notable successes in NVS tasks. A diverse range of neural representations, including MLPs [5], [10]–[12], voxel grids [15]–[17], tri-planes [18], [19], vector decomposition [14], and multi-level hash grids [13], have been extensively explored for both reconstruction and synthesis. However, the majority of these studies are concentrated on object-centric reconstructions of small indoor scenes. More recently, several works [7], [20]–[25] have gradually expanded these approaches to large-scale outdoor environments. However, most existing approaches focus on either the camera or LiDAR domain, with only a few [26], [27] addressing both modalities concurrently. A significant challenge remains in handling both static and dynamic scenes, a crucial requirement for real-world driving scenarios. This limitation impacts the scalability of such networks. In this paper, we propose a unified framework that offers NVS for both LiDAR and camera.

(b) *NeRF for LiDAR*: Traditional NeRF models were initially designed for camera images and cannot be directly applied to LiDAR point clouds due to the fundamental structural differences between these two data types. Unlike images, which capture dense 2D pixel grids, LiDAR provides sparse, 3D point cloud data, making it challenging to adapt NeRF for LiDAR's spatial properties. Several efforts have been made to extend NeRF to the LiDAR domain. Early notable works include Neural LiDAR Fields for Novel view synthesis [28] and LiDAR-NeRF [22], which can synthesise novel synthesis of LiDAR point cloud. These were followed by additional research, such as NeRF-LiDAR [29], NeuRAD [26] and LiDAR4D [7]. Among these, NeRF-LiDAR [29] and NeuRAD [26] stand out for utilizing both LiDAR and camera data jointly to learn NeRF fields, enhancing their ability to generate more accurate and detailed reconstructions by leveraging the complementary nature of both sensor modalities. The integration of LiDAR and camera data mitigates the inherent limitations associated with single-modal approaches, thereby enabling more robust and effective neural radiance field implementations in complex 3D environments. However, both of these models utilise 3D annotation of LiDAR point cloud for tracking moving objects. As data annotation is an expensive process therefore,

it's not always possible to invest time and resources in it. In our work, we are learning dynamic scene content in a self-supervised manner by employing the scene flow module.

## III. METHOD

The network of the proposed framework is inspired by existing works such as NeRF [5], LiDAR-NeRF [22], K-Planes [8], LiDAR4D [7] and Neural Scene Flow Prior [30]. A novel contribution of the proposed network includes multimodal training by using both LiDAR and camera data for jointly learning NeRF models of LiDAR and camera, heuristic image pixel sampling, and improved gradient loss for better feature alignment. Figure 2 illustrates the proposed network; it comprises of five individual MLPs and two parameterised 4D hybrid encodings using Multi-resolution Hash encoding [13] and K-Planes encoding [8]. Each module of the proposed network serves a different learning objective such as: (1) Scene Flow MLP learns the forward and backward scene flow; (2) Positional, Spherical Harmonics and Multi-Resolution Hash encoding encodes positional and temporal information into high-dimensional feature vectors; (3) 4D Hybrid encoding encodes the space and time information using multi-resolution hash and 2D-planes for the decomposition of static and dynamic contents of a scene; (4) NeRF MLP learn the implicit neural representation of 4D scene in spatio-temporal coordinate frame; (5) Three radiance MLP handles the view-dependent attributes such as raydrop, intensity of LiDAR point cloud and RGB pixel value of camera image; (6) U-Net is used for global refinement of raydrop for better consistency of synthesised point cloud throughout the scene.

By combining all modules, there are around 132 million trainable parameters to be optimised during network training. Unlike recent research on NeRF-based multimodal frameworks for NVS such as MARS [31], NeRF-LiDAR [23] and NeuRAD [26], the proposed network decouples static (background) and dynamic (foreground) scene components without any external supervision. All modules work coherently to synthesise views of LiDAR and Camera with high fidelity for large-scale static and dynamic scenarios of autonomous vehicle. Therefore, the proposed NVSF of LiDAR and camera can be formalised as a function of LiDAR  $(\sigma, \mathbf{i}, \mathbf{p}) = f_{lidar}(\mathbf{x}, \theta)$  and Camera  $(\sigma, \mathbf{c}) = f_{camera}(\mathbf{x}, \theta)$ . Where  $\mathbf{i}$ ,  $\mathbf{p}$ ,  $\mathbf{c}$  denotes lidar intensity, ray-drop probabilities and camera RGB values respectively.

### A. Geometric Feature Alignment

LiDAR's uniform laser projection creates characteristic patterns in point clouds, particularly in flat regions, such as road surface, building walls and car bonnet. The vanilla NeRF struggles with capturing these local features as it's optimised with global pano image objective. As a result, despite performing well on PCD metrics such as Chamfer distance (CD), it struggles with Peak Signal to noise ratio (PSNR) score. LiDAR-NeRF [22] proposed structural regularization loss, to improve local geometric consistency.

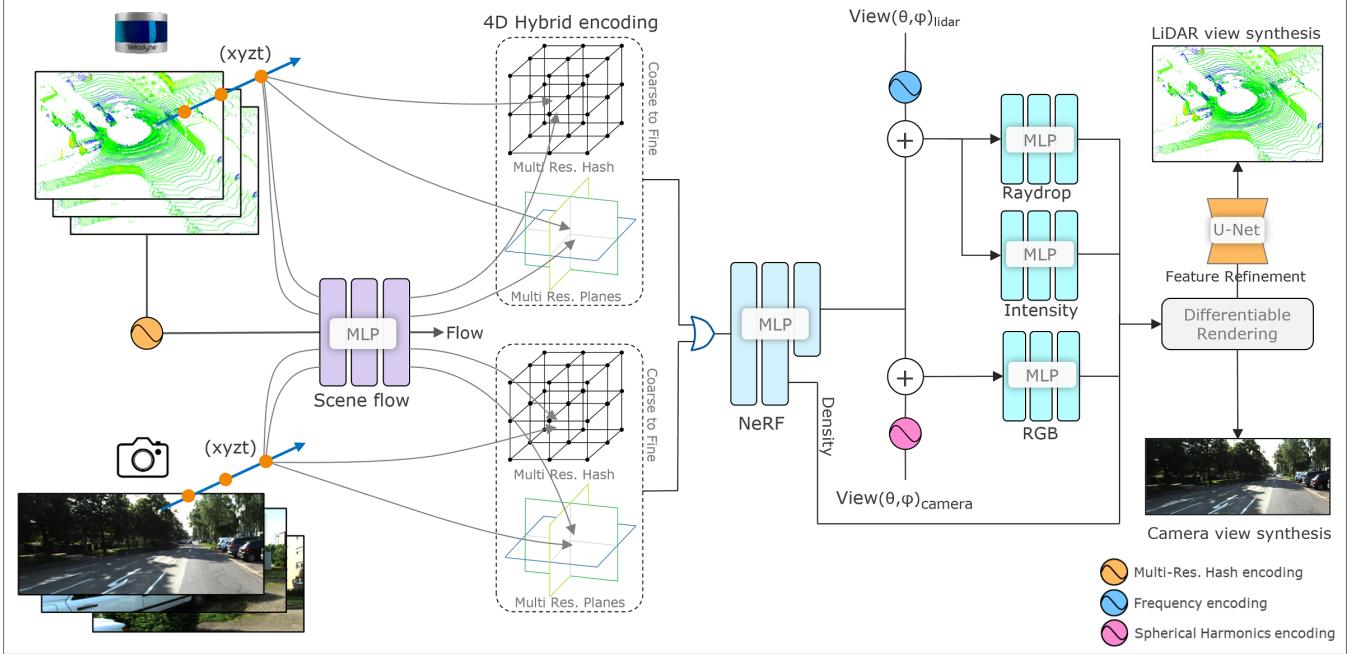


Fig. 2: The proposed network consists of Scene flow MLP for learning the temporal features, 4D hybrid hash-planes encoding for the encoding of space and time coordinates, and a shared MLP for learning the scene density and global features. A differentiable rendering module for calculating depth, intensity, raydrop of LiDAR point cloud and pixel colour. U-Net network is used for global refinement of LiDAR renderings, as mentioned in LiDAR4D [7].

To compute targeted areas for applying gradient-based loss, single gradients of the ground truth panoramic depth image were used to create the clipping mask. A threshold was applied to these gradients to control the coverage of the gradient loss through the mask. However, analysis revealed that the single gradient-based clipping mask failed to capture distant regions of the point cloud, as illustrated in Figure 3a. Increasing the threshold for the clipping mask led to over-smoothing of the panoramic depth image. As gradient loss was applied to non-flat areas, reducing the accuracy of the synthesised point cloud. This issue also affects LiDAR4D [7], which uses the same gradient loss function as evident in their codebase.

Unlike single gradients, double gradients of pano depth image of far away regions such as ground and walls of the scene are zero. Gradient loss can be computed on a large relevant area of point cloud by using a small threshold (eg: 0.01) on pano image, as shown in Figure 3b. This method covers both sparse-texture (far distance) and dense-texture (near distance) regions of the point cloud. Figure 3b shows the points where Gradient loss is computed. The double gradient mask is better as it can cover more (black region) geometrically structured regions for feature alignment.

Gradient loss through a Double gradient-based clipping mask is calculated at small cropped regions (patches) of the pano depth image. Gradient loss of both  $x$  and  $y$  directions are calculated and combined gradient loss for  $x$  and  $y$  directions are optimised in local patches. Loss is given by the Equation 1, where  $R$  is the set of training rays of local patches, and  $G_M(\cdot)$  denotes the gradient operation of  $x$  and

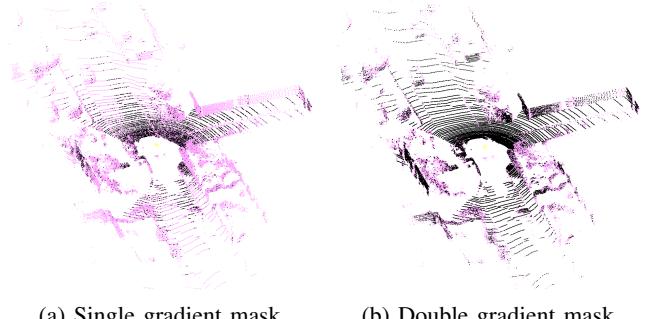


Fig. 3: Compared to a single-gradient approach (a), our method (b) captures sparse features. The black regions are chosen for structural regularization, while the pink-highlighted areas are skipped due to their lack of geometric features.

$y$  directions. The loss equation is derived from LiDAR-NeRF [22] structural regularization.

$$\mathcal{L}_{\text{grad}} = \left\| \hat{G}_M(\mathbf{R}) - G_M(\mathbf{R}) \right\|_1 \quad (1)$$

### B. Heuristic Ray Sampling

Naive NeRF-based models originally used a uniform ray sampling approach, this method assumed that each pixel contained the same amount of information. However, this assumption fails in large, unbounded scenes, such as those in autonomous driving, where information entropy is non uniform. Failure to consider this leads to NeRF rendering

ghost artifacts, especially around complex objects such as, pedestrians. To address this, strategy of Non-uniform sampling was introduced in [32], but it was targeted for the small indoor 360° scenes. The NVSF employs heuristic-based pixel sampling during training, guided by a multinomial probability distribution. This content-aware ray sampling adapts to pixel-wise reconstruction loss, giving higher sampling probabilities to pixels with higher loss. As illustrated in Figure 4, this method draws more pixels with high reconstruction loss compared to random sampling. Since all pixels have a positive likelihood of being sampled, this approach balances exploration and exploitation by focusing on both high-loss and unexplored regions. To introduce variability, sampled pixel indices are perturbed through jittering. The framework alternates between heuristic and random sampling across training epochs, allowing the network to explore new pixels in one epoch and minimize reconstruction loss in the next, thereby improving focus on challenging pixels.

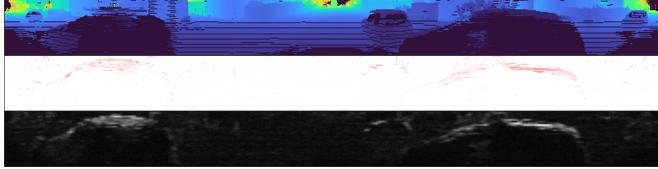


Fig. 4: Figure shows the Heuristic based pixel sampling of LiDAR’s Pano image. From top to bottom: Pano depth, Reconstruction loss and Heuristic pixel sampling. Red pixels in the middle image indicate high reconstruction loss, thus increasing their sampling probability, shown as white pixels in the bottom image.

### C. Multimodal NeRF

Depth supervision from LiDAR has already shown to improve image based NeRF [33], similarly images complement the sparse LiDAR data. Building on this NeRF-LiDAR [29] and NeuRAD [26] also use multimodal data for NVS of both sensors. But these networks require 3D annotations for tackling dynamic problem, which involves additional labelling effort. The proposed framework can do NVS for both sensors in a self-supervised manner by employing a 3D flow network. A strategy is required to fuse encoded features effectively for improved convergence of the proposed network. Recent research [34] indicates that different modalities often contradict each other due to spatial and temporal misalignment. Moreover, LiDAR and camera modalities exhibit inconsistent representations and converge at different rates, leading to uncoordinated convergence issues [35], [36], [37]. We conduct extensive experiments as shown in Table I with various combinations of encoded features from both modalities:

- 1) Common encoding for Camera and LiDAR
- 2) Separate encoding for Camera and LiDAR
- 3) Separate encoding for Camera and LiDAR with implicit fusion before passing to NeRF MLP

Separate 4D Hybrid Hash-Planar encoding for LiDAR and Camera data performed best among all the three options

Strategy	LiDAR			Camera	
	PCD	Range			
	CD ↓	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑
Same Enc.	0.175	0.76	27.12	0.712	24.97
Separate Enc.	<b>0.110</b>	<b>0.792</b>	<b>27.85</b>	<b>0.811</b>	<b>25.21</b>
Separate Enc.+F.	0.161	0.757	27.52	0.809	25.14

TABLE I: Comparison of different encoding strategies.

### D. Implementation

We follow the classical differential rendering approach used in NeRF models [5], [7], [22]. The view direction of rays is similar to the pattern of a spinning mechanical LiDAR, whose beams are cast in 360°. The expected depth  $\hat{D}(\mathbf{r})$  of a beam can be obtained by integrating over samples given by equation 2, which is derived from LiDAR4D [7].

$$\hat{D}(\mathbf{r}) = \sum_{i=1}^N T_i (1 - e^{-\sigma_i \delta_i}) t_i, \quad \alpha_i = 1 - e^{-\sigma_i \delta_i} \quad (2)$$

Where  $T_i = e^{(-\sum_{j=1}^{i-1} \sigma_j \delta_j)}$  indicates the accumulated transmittance along ray  $\mathbf{r}$  to the sampled point  $t_i$ , and  $\delta_i = t_{i+1} - t_i$  is the distance between adjacent samples.  $t_i$  is the depth value of queried points on the ray  $r$ , and  $\alpha_i$  is the opacity value. The expected depth represents the distance from the LiDAR sensor. Moreover, both the origin and viewing direction of the ray are transformed to the global world coordinate system, allowing the sampled points  $t_i$  to be actual points in the real world and consistent across multiple frames (pano image). The view dependent attributes for LiDAR are same as in LiDAR4D [7] and for camera attributes are calculated as per Image NeRF [5].

As vehicle motion range may span a long distance in autonomous driving scenarios, it is extremely difficult to establish long-term correspondences in the canonical space without a scene flow network. A few models for the prediction of points translation in a space-time volume are Neural scene flow prior [38], Fast Neural Scene Flow (FastNSF) [39], NeuralPCI [40], Dynibar [41] and LiDAR4D [7]. We utilise the LiDAR4D scene flow network to enhance temporal consistency in spatio-temporal coordinates via a 4D hybrid multi-resolution hash and plane encoding. This network predicts the prior and post positions of points between adjacent frames and aggregates multi-frame dynamic features for time-consistent reconstruction. During Flow MLP training, forward and backward scene flow is predicted using the prior and subsequent frames, capturing changes in point positions across frames. CD is used for calculating geometric loss ( $\mathcal{L}_{flow}$ ) between the prediction points  $\hat{S}$  and ground truth  $S$  points of the current frame as given by equations 3 and 4. It encourages the two adjacent frames of the point cloud transformed by the predicted scene flow to be as consistent as possible.

$$CD = \frac{1}{K} \sum_{\hat{p}_i \in \hat{S}} \min_{p_i \in S} \|\hat{p}_i - p_i\|_2^2 + \frac{1}{K} \sum_{p_i \in S} \min_{\hat{p}_i \in \hat{S}} \|\hat{p}_i - p_i\|_2^2 \quad (3)$$

$$\mathcal{L}_{\text{flow}} = \sum_{j \in \pm 1} \text{CD}(S_i + \text{MLP}_{\text{flow}}(S_i), S_{i+j}), i \in (0, n-1) \quad (4)$$

Total reconstruction of the proposed framework is a combination of weighted six optimization objectives as shown in the Equation 9. The  $\mathcal{L}_{\text{rgb}}$  is derived from Image NeRF [5] and the rest of the loss function is a combination of LiDAR4D [7] and LiDAR-NeRF [22].

$$\mathcal{L}_{\text{depth}} = \sum_{\mathbf{r} \in R} \left\| \hat{D}(\mathbf{r}) - D(\mathbf{r}) \right\|_1 \quad (5)$$

$$\mathcal{L}_{\text{intensity}} = \sum_{\mathbf{r} \in R} \left\| \hat{I}(\mathbf{r}) - I(\mathbf{r}) \right\|_2^2 \quad (6)$$

$$\mathcal{L}_{\text{raydrop}} = \sum_{\mathbf{r} \in R} \left\| \hat{P}(\mathbf{r}) - P(\mathbf{r}) \right\|_2^2 \quad (7)$$

$$\mathcal{L}_{\text{rgb}} = \sum_{\mathbf{r} \in R} \left\| \hat{C}(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \quad (8)$$

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{flow}} + \lambda_2 \mathcal{L}_{\text{depth}}(\mathbf{r}) + \lambda_3 \mathcal{L}_{\text{intensity}}(\mathbf{r}) \quad (9)$$

$$+ \lambda_4 \mathcal{L}_{\text{raydrop}}(\mathbf{r}) + \lambda_5 \mathcal{L}_{\text{grad}} + \lambda_6 \mathcal{L}_{\text{rgb}}$$

where  $\mathcal{L}_{\text{depth}}$ ,  $\mathcal{L}_{\text{intensity}}$ ,  $\mathcal{L}_{\text{raydrop}}$  and  $\mathcal{L}_{\text{rgb}}$  are losses for Depth, Intensity, Raydrop and RGB of a corresponding pixel. Scene flow loss  $\mathcal{L}_{\text{flow}}$  given by the Equation 4.  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$ ,  $\lambda_4$ ,  $\lambda_5$  and  $\lambda_6$  are their corresponding weights. Mean Absolute Error is used for calculating Depth loss while Mean Square Error is used for calculating Intensity loss, Raydrop loss, Gradient loss and RGB loss.

#### IV. EXPERIMENTS

The proposed NVSF is implemented based on LiDAR-NeRF [22] and LiDAR4D [7] codebase. The framework is trained on the KITTI-360 [42] dataset using its Velodyne HDL-64E LiDAR, two front wide-angle cameras and ego position data. We train on 4 static and 6 dynamic scenes (same as mentioned in LiDAR4D) each comprising 64 individual frames. Out of this, 60 frames are used for training, and 4 frames are used for validation purposes. The height and width of the LiDAR pano image is set to 66 and 1030. Before training, LiDAR point clouds are centered and scaled such that the region of interest falls within a unit cube. All frames are randomly sampled to prevent over-fitting and promote consistent learning. A ray is sampled uniformly at 768 points in Axis Aligned Bounding Box (AABB) frame. The optimization of NVSF network is implemented in PyTorch with Adam [43] optimiser. The maximum iterations are set to 30k for each scene, with a batch size of 2048 rays per an image. Learning rate is 1e-2 with exponential decay during iterations and a final decay coefficient of 0.1. The values of weights  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$ ,  $\lambda_4$ ,  $\lambda_5$  and  $\lambda_6$  are 1.0, 1.0, 0.1, 0.01, 0.01 and 1.0 respectively. Remaining parameters such as 4D-Hybrid encoding and U-Net are kept same as the LiDAR4D network.

#### A. Baseline

For a thorough evaluation of the NVS framework, a comprehensive comparison is conducted with LiDAR-NeRF [22] and LiDAR4D [7] models on both static and dynamic scenes reconstruction. LiDAR4D and LiDAR-NeRF has demonstrated superior performance compared to other LiDAR-based NVS methods such as LiDARSim [3] and PCGen [44]. Given their proven efficacy, it is deemed unnecessary to include these other networks in the comparison. The baseline models are trained using their available GitHub code. The primary focus of this work is to synthesise multimodal data though unified framework and introduce novel contributions to enhance their performance.

Modules	LiDAR (Foreground)			
	PCD	Range RMSE ↓	Intensity RMSE ↓	Raydrop RMSE ↓
<b>Baseline</b>	0.220	0.968	0.025	0.093
+ Heuristic Sampling	0.173	0.846	0.023	0.080
+ Feature alignment	0.158	<b>0.813</b>	0.023	<u>0.079</u>
+ Multimodal	0.137	0.833	0.024	0.081
<b>NVSF</b>	<b>0.128</b>	<u>0.818</u>	<b>0.023</b>	<b>0.078</b>

TABLE II: Comparison of LiDAR based metrics for different modules. The ablation is conducted over a subset of scenes.

#### B. LiDAR Evaluation

We report evaluation metrics for different attributes such as Point Cloud, Range, Intensity, Raydrop and camera image quality. The metrics are averaged over the 10 scenes, as mentioned above. To analyse dynamic point reconstruction, we report these metrics for the foreground section too. 3D annotation of KITTI360 dataset [42] is used for the foreground separation. The proposed NVS framework outperformed both LiDAR-NeRF [22] and LiDAR4D [7] models in almost every PCD synthesis metric. This difference is especially highlighted in the foreground section, as reported in Table III. To analyse the effectiveness of individual contributions, we conduct an ablation study as shown in Table II. Maximum improvement is shown in the feature alignment, and the efficacy of the multimodal fusion is also evident. Our method has shown that multimodal NVS can have a shared NeRF without impacting individual performance. The framework has shown improvements of 29% in CD and 12% in Raydrop Root Mean Square Error (RMSE). Higher values of PSNR suggest that the synthesised point cloud has less noise compared to both baseline models.

The proposed network has accurately synthesised background and foreground objects with view-dependent attributes compared to baseline and LiDAR4D [7] networks, as shown in Figure 5. Its PCD has better feature representation, consistency and high fidelity.

#### C. Camera Evaluation

The proposed NVS framework uses multimodal data from LiDAR and camera for training the network, thereby jointly learning implicit Neural scene representation and view-dependent attributes. The framework can synthesise camera

Method	LiDAR												
	PCD		Range				Intensity				Raydrop		
	CD ↓	F-score ↑	RMSE ↓	LPIPS ↓	SSIM ↑	PSNR ↑	RMSE ↓	LPIPS ↓	SSIM ↑	PSNR ↑	RMSE ↓	Acc.↑	F-score ↑
<b>Static + Dynamic Scene</b>													
LiDAR-Nerf [22]	0.114	<u>0.918</u>	3.982	0.309	0.623	26.208	0.152	0.363	0.325	16.442	0.356	0.832	0.882
LiDAR4D [7]	0.129	0.905	<u>3.240</u>	<u>0.108</u>	<u>0.796</u>	<u>28.045</u>	<u>0.117</u>	<u>0.193</u>	<u>0.564</u>	<u>18.667</u>	<u>0.274</u>	<u>0.921</u>	<u>0.937</u>
NVSF	<b>0.091</b>	<b>0.929</b>	<b>3.150</b>	<b>0.091</b>	<b>0.819</b>	<b>28.322</b>	<b>0.115</b>	<b>0.163</b>	<b>0.579</b>	<b>18.862</b>	<b>0.240</b>	<b>0.925</b>	<b>0.947</b>
<b>Dynamic Scene (Foreground)</b>													
LiDAR-Nerf [22]	0.758	0.408	0.799	0.053	0.967	40.885	0.025	0.033	<u>0.973</u>	32.941	0.090	0.988	0.411
LiDAR4D [7]	<u>0.255</u>	<u>0.501</u>	<u>0.726</u>	<u>0.039</u>	<u>0.973</u>	<u>41.436</u>	<u>0.021</u>	<u>0.026</u>	<b>0.979</b>	<u>34.314</u>	<u>0.076</u>	<u>0.991</u>	<u>0.619</u>
NVSF	<b>0.111</b>	<b>0.720</b>	<b>0.645</b>	<b>0.030</b>	<b>0.976</b>	<b>42.459</b>	<b>0.020</b>	<b>0.019</b>	<b>0.979</b>	<b>34.870</b>	<b>0.069</b>	<b>0.992</b>	<b>0.733</b>

TABLE III: Quantitative comparison on KITTI 360 Dataset. We report LiDAR based metrics for Static, Dynamic and Foreground (dynamic lidar Points) scenes. **Bold** and underlined values are best and second best respectively for the corresponding metrics.

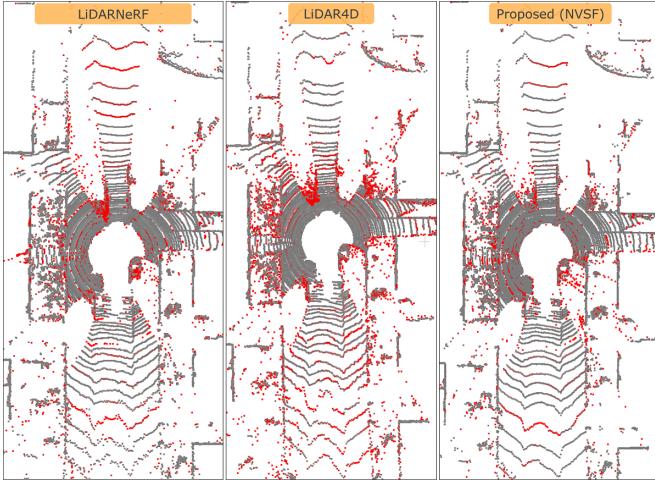


Fig. 5: Comparison of error map of synthesised PCD by LiDAR-NeRF, LiDAR4D and Proposed NVSF for dynamic sequence 4950. Red colour points are indicating an error of more than 1.0m with respect to ground truth.

images along with LiDAR point clouds. Since neither of the Baseline networks support joint synthesis of LiDAR and Camera data, we only report evaluation of our work. Table IV shows the image quality metrics for NVSF model. The combined PSNR value is low when compared to foreground section, this could be due to the presence of limited dynamic objects. To give a perspective on our quality of image synthesis, NeuRad [26] (supervised approach) has reported a PSNR value of 27 for KITTI MOT dataset [45]. This indicates the effectiveness of self-supervised approach for dynamic object view synthesis.

Scenes	Camera			
	RMSE ↓	LPIPS ↓	SSIM ↑	PSNR ↑
Static + Dynamic	1.305	0.217	0.817	25.432
Dynamic (Foreground)	0.344	0.017	0.736	49.057

TABLE IV: Table shows the evaluation result for the reconstruction of the Camera image for KITTI 360 Dataset.

## V. CONCLUSION

This paper has proposed a NVS framework for synthesis of both LiDAR and Camera data by learning implicit Neural representation of a 4D scene and view-dependent attributes in a canonical frame. We report a best performance on quantitative and qualitative metrics when compared to baselines, LiDAR-NeRF and LiDAR4D. We revisit the fundamentals of ray sampling and improve upon this by introducing Heuristic sampling and Double gradient mask. The proposed network is able to synthesise both static and dynamic scenes in a self-supervised manner. Despite the performance, the current self-supervision approach for dynamic scene reconstruction binds foreground and background scenes, which restrict NVS within the ego-vehicle trajectory in the spatio-temporal coordinate space. Consequently, scene editing or scene manipulation is not possible outside ego-vehicle trajectory. We foresee the future multimodal work will focus on improving the reconstruction quality over larger scenes without 3D annotation supervision.

## VI. ACKNOWLEDGEMENT

This work is a part of a master thesis supported by AVL Software and Functions GmbH, Germany and Technische Hochschule Deggendorf, Germany. We are thankful to publishers of LiDAR-NeRF and LiDAR4D papers for their great work and making the codebase publicly available for further research.

## REFERENCES

- [1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Conference on robot learning*. PMLR, 2017, pp. 1–16.
- [2] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles,” in *Field and Service Robotics: Results of the 11th International Conference*. Springer, 2018, pp. 621–635.
- [3] S. Manivasagam, S. Wang, K. Wong, W. Zeng, M. Sazanovich, S. Tan, B. Yang, W.-C. Ma, and R. Urtasun, “LiDARsim: Realistic LiDAR simulation by leveraging the real world,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 167–11 176.
- [4] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pee, “BlenSor: Blender sensor simulation toolbox,” in *Advances in Visual Computing: 7th International Symposium, ISVC 2011, Las Vegas, NV, USA, September 26–28, 2011. Proceedings, Part II 7*. Springer, 2011, pp. 199–208.
- [5] 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.
- [6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [7] Z. Zheng, F. Lu, W. Xue, G. Chen, and C. Jiang, “LiDAR4D: Dynamic Neural Fields for Novel Space-time View LiDAR Synthesis,” *arXiv preprint arXiv:2404.02742*, 2024.
- [8] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, “K-planes: Explicit radiance fields in space, time, and appearance,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 479–12 488.
- [9] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide, “Neural Scene Graphs for Dynamic Scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2856–2865.
- [10] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.
- [11] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, “Mip-NeRF 360: Unbounded anti-aliased neural radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.
- [12] ———, “Zip-NeRF: Anti-aliased grid-based neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 697–19 705.
- [13] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.
- [14] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, “Tensorf: Tensorial radiance fields,” in *European conference on computer vision*. Springer, 2022, pp. 333–350.
- [15] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, “Plenoxels: Radiance fields without neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5501–5510.
- [16] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, “Neural sparse voxel fields,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020.
- [17] C. Sun, M. Sun, and H.-T. Chen, “Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5459–5469.
- [18] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis, et al., “Efficient geometry-aware 3d generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 123–16 133.
- [19] W. Hu, Y. Wang, L. Ma, B. Yang, L. Gao, X. Liu, and Y. Ma, “Tri-MipRF: Tri-Mip representation for efficient anti-aliasing neural radiance fields,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 774–19 783.
- [20] S. Sun, B. Zhuang, Z. Jiang, B. Liu, X. Xie, and M. Chandraker, “LidaRF: Delving into LiDAR for neural radiance field on street scenes,” *arXiv preprint arXiv:2405.00900*, 2024.
- [21] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, “Block-NeRF: Scalable large scene neural view synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8248–8258.
- [22] T. Tao, L. Gao, G. Wang, Y. Lao, P. Chen, H. Zhao, D. Hao, X. Liang, M. Salzmann, and K. Yu, “LiDAR-NeRF: Novel LiDAR view synthesis via neural radiance fields,” *arXiv preprint arXiv:2304.10406*, 2023.
- [23] J. Zhang, F. Zhang, S. Kuang, and L. Zhang, “NeRF-LiDAR: Generating Realistic LiDAR Point Clouds with Neural Radiance Fields,” in *AAAI Conference on Artificial Intelligence (AAAI)*, 2024.
- [24] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, “Urban radiance fields,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 932–12 942.
- [25] Z. Wang, T. Shen, J. Gao, S. Huang, J. Munkberg, J. Hasselgren, Z. Gojcic, W. Chen, and S. Fidler, “Neural fields meet explicit geometric representations for inverse rendering of urban scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8370–8380.
- [26] A. Tonderski, C. Lindström, G. Hess, W. Ljungbergh, L. Svensson, and C. Petersson, “NeuRAD: Neural rendering for autonomous driving,” *arXiv preprint arXiv:2311.15260*, 2023.
- [27] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun, “UniSim: A Neural Closed-Loop Sensor Simulator,” in *CVPR*, 2023.
- [28] S. Huang, Z. Gojcic, Z. Wang, F. Williams, Y. Kasten, S. Fidler, K. Schindler, and O. Litany, “Neural LiDAR fields for novel view synthesis,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 236–18 246.
- [29] J. Zhang, F. Zhang, S. Kuang, and L. Zhang, “NeRF-LiDAR: Generating realistic LiDAR point clouds with neural radiance fields,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 7178–7186.
- [30] X. Li, J. Kaesemodel Pontes, and S. Lucey, “Neural Scene Flow Prior,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 7838–7851. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2021/file/41263b9a46f6f8f22668476661614478-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/41263b9a46f6f8f22668476661614478-Paper.pdf)
- [31] Z. Wu, T. Liu, L. Luo, Z. Zhong, J. Chen, H. Xiao, C. Hou, H. Lou, Y. Chen, R. Yang, et al., “MARS: An instance-aware, modular and realistic simulator for autonomous driving,” in *CAAI International Conference on Artificial Intelligence*. Springer, 2023, pp. 3–15.
- [32] T. Otonari, S. Ikehata, and K. Aizawa, “Non-uniform sampling strategies for NeRF on 360° images,” in *BMVC*, 2022, p. 344.
- [33] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, “Depth-supervised NeRF: Fewer Views and Faster Training for Free,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 872–12 881.
- [34] T. Tang, G. Wang, Y. Lao, P. Chen, J. Liu, L. Lin, K. Yu, and X. Liang, “AlignMiF: Geometry-Aligned Multimodal Implicit Field for LiDAR-Camera Joint Synthesis,” *arXiv preprint arXiv:2402.17483*, 2024.
- [35] X. Peng, Y. Wei, A. Deng, D. Wang, and D. Hu, “Balanced multimodal learning via on-the-fly gradient modulation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8238–8247.
- [36] Y. Sun, S. Mai, and H. Hu, “Learning to balance the learning rates between various modalities via adaptive tracking factor,” *IEEE Signal Processing Letters*, vol. 28, pp. 1650–1654, 2021.
- [37] W. Wang, D. Tran, and M. Feiszli, “What makes training multimodal classification networks hard?” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 695–12 705.
- [38] X. Li, J. Kaesemodel Pontes, and S. Lucey, “Neural scene flow prior,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 7838–7851, 2021.
- [39] X. Li, J. Zheng, F. Ferroni, J. K. Pontes, and S. Lucey, “Fast Neural Scene Flow,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 9878–9890.

- [40] Z. Zheng, D. Wu, R. Lu, F. Lu, G. Chen, and C. Jiang, “NeuralPCI: Spatio-temporal Neural Field for 3D Point Cloud Multi-frame Non-linear Interpolation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 909–918.
- [41] Z. Li, Q. Wang, F. Cole, R. Tucker, and N. Snavely, “DynIBaR: Neural dynamic image-based rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4273–4284.
- [42] Y. Liao, J. Xie, and A. Geiger, “KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2022.
- [43] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [44] G. A. Allen and L. M. Baldwin, “PCGen: A FORTRAN IV program to generate paired-comparison stimuli,” *Behavior Research Methods & Instrumentation*, vol. 12, no. 3, pp. 383–384, 1980.
- [45] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, “MOTS: Multi-Object Tracking and Segmentation,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.