

SLCF-Net: Sequential LiDAR-Camera Fusion for Semantic Scene Completion using a 3D Recurrent U-Net

Helin Cao and Sven Behnke

Abstract— We introduce SLCF-Net, a novel approach for the Semantic Scene Completion (SSC) task that sequentially fuses LiDAR and camera data. It jointly estimates missing geometry and semantics in a scene from sequences of RGB images and sparse LiDAR measurements. The images are semantically segmented by a pre-trained 2D U-Net and a dense depth prior is estimated from a depth-conditioned pipeline fueled by Depth Anything. To associate the 2D image features with the 3D scene volume, we introduce Gaussian-decay Depth-prior Projection (GDP). This module projects the 2D features into the 3D volume along the line of sight with a Gaussian-decay function, centered around the depth prior. Volumetric semantics is computed by a 3D U-Net. We propagate the hidden 3D U-Net state using the sensor motion and design a novel loss to ensure temporal consistency. We evaluate our approach on the SemanticKITTI dataset and compare it with leading SSC approaches. The SLCF-Net excels in all SSC metrics and shows great temporal consistency.

I. INTRODUCTION

3D Semantic Scene Completion (SSC) aims to simultaneously estimate the complete geometry and semantics of a scene from sensor data, a task that has garnered increased attention in the computer vision and robotics community. Many existing methods still rely on single sensor input, such as RGB images or depth data (e.g., occupancy grids, point clouds, depth maps, etc.). Both RGB images and depth data can provide valuable information about the environment. However, intuitively, RGB images enable a dense interpretation of the semantic content with high spatial resolution, while depth data provides the scene geometry. The complementary nature of these two data modalities can facilitate SSC. Although RGB-D cameras directly provide both color and depth information, their short depth range (typically less than 5 m) limits their application in outdoor urban scenarios. For autonomous vehicles, a common alternative is the combination of RGB images and LiDAR scans [1], [2]. Addressing this prevalent configuration, we propose a novel framework that leverages both RGB and LiDAR data.

In this work, we introduce SLCF-Net, designed to fuse sequences of RGB images and sparse 3D LiDAR scans to infer a 3D voxelized semantic scene. Fig. 1 illustrates the inputs and outputs for urban driving scenarios. The RGB image, sourced from the KITTI dataset [1], is paired with a sparse depth map, projected from a calibrated LiDAR scan.

This research has been supported by MBZIRC prize money. All authors are with the Autonomous Intelligent Systems group, Computer Science Institute VI – Intelligent Systems and Robotics – and the Center for Robotics and the Lamarr Institute for Machine Learning and Artificial Intelligence, University of Bonn, Germany; caoh@ais.uni-bonn.de

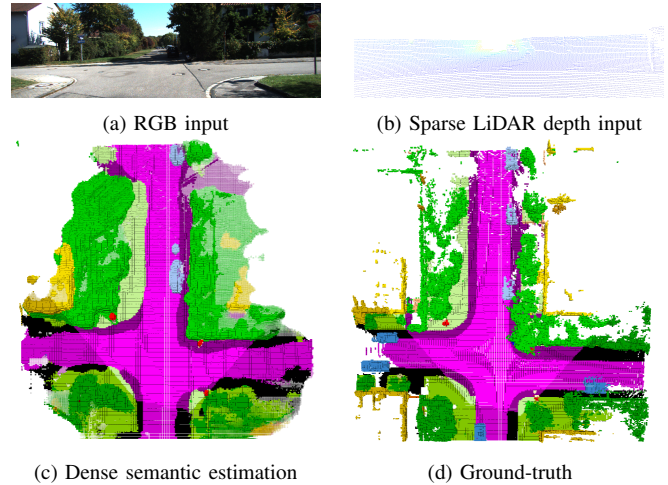


Fig. 1: SLCF-Net estimates the dense semantic scene as shown in (c) using sequences of RGB images (a) and aligned sparse LiDAR depth maps (b). Both (c) and (d) depict a voxelized scene as defined by the SemanticKITTI Benchmark [3] from the bird’s-eye view. Parts of the scene in both the estimation (c) and the ground-truth (d) lie outside of the field of view (FoV), which are visualized as shadow areas. The unknown areas, as defined by the ground truth, are visualized at 20% opacity in (c).

The SLCF-Net then recursively estimates dense occupancy and semantic labels within the 3D scene volume.

Lifting a 3D scene from a 2D image is a well-known ill-posed problem [4]. However, depth values provide crucial priors during scene reconstruction. To enhance the density of the depth map, we utilized the power of Depth Anything Model [5], which densely estimates relative distance from an RGB image. Then we scale the relative distance based on the raw sparse depth input to obtain the dense and absolute depth estimation. The dense depth estimation is utilized to project 2D features into the 3D volume. To model sensor noise and inference uncertainty, we smooth the projection with a depth-dependent function. Along the line of sight, as the distance from the depth prior increases, the weight associated with a voxel diminishes. This decrease is effectively captured using a Gaussian-decay function.

To the best of our knowledge, existing SSC methods reconstruct the scene from data in the current frame without considering historical information. Considering the scanning process in an autonomous vehicle, there is a large range of overlapping volumes between successive frames. Thus, the SSC model can benefit from leveraging previous frames. With the known poses of the frames, we propagate the 3D semantic scene representation from previous frames to compensate for the sensor motion. Benefiting from the ability of

CNNs to preserve the spatial ordering of data, this alignment can be utilized not only in the output layer but also for feature fusion in latent space. We store the propagated previous feature as the hidden state and concatenate it with the current feature. We also designed an inter-frame consistency loss to enforce temporal consistency in the overlap area.

In summary, the main contributions of this paper include:

- introduction of SLCF-Net: a novel SSC method that fuses 2.5D sparse depth maps with 2D RGB images, adaptable to various sensor configurations;
- the GDP module: a Gaussian-decay Depth-prior Projection method for projecting 2D features into 3D;
- a mechanism that propagates features from the previous frame to the current frame based on the known coordinate transformation.

II. RELATED WORK

A. 3D Semantic Scene Completion (SSC)

Research on inferring missing geometry based on existing information has spanned decades. Traditional approaches typically involved filling small holes by employing continuous energy minimization, such as Poisson surface reconstruction [6], [7]. However, the advent of deep neural networks marked a paradigm shift. SSCNet [8], for instance, became the first to jointly infer an entire scene’s geometry and semantics from a monocular depth image, defining the ‘SSC’ task as requiring joint inference of geometry and semantics. Over the years, various data types have been employed as inputs for SSC methods, as meticulously documented in a survey by Roldao et al. [9]. Most current works utilize inputs such as depth [10], occupancy grids [11], [12], point clouds [13]–[15], and RGB [16]–[18]. In this work, we address SSC by utilizing a unique combination of inputs: an RGB image coupled with a sparse depth map—an approach particularly tailored for the sensor configuration of autonomous vehicles.

B. Sensor Fusion

Sensor fusion is a technique that combines data from diverse sensory sources to enhance the perception and understanding of the environment. Numerous studies focus on the fusion of camera and LiDAR data to achieve refined 3D reconstructions or precise pose estimations. Riegler et al. [19] proposed a method to combine the high-resolution color information from cameras with LiDAR depth data using a voxel-based format, improving 3D reconstruction accuracy. Czarnowski et al. [20] introduced a keyframe-based SLAM method that fuses camera and LiDAR for robust pose estimation in dynamic environments. Bultmann et al. addressed multiple tasks by the fusion of smart edge sensors with overlapping fields of view, like human pose tracking [21], robot pose estimation [22], and dynamic object tracking [23]. They also introduced a UAV system tailored for real-time semantic interpretation, integrating multiple sensor modalities such as LiDAR, RGB-D camera, and thermal camera [24].

C. Sequence Learning

Sequence learning involves learning from previous elements to support the understanding of current elements or the prediction of future elements. Sequence learning can help the model produce results based on historical information. In SSC, this approach will help improve temporal consistency. Historically, this field was primarily centered around machine translation. Classical architectures like RNN [25], GRU [26], and LSTM [27] emerged during this early era. In recent years, sequence learning expanded to video understanding. Simonyan et al. [28] presents a two-stream network that divides the video into spatial and temporal streams. While the spatial stream focuses on individual frame processing to capture appearance attributes, the temporal stream interprets optical flow fields between frames, targeting motion dynamics comprehension. Villar-Corrales et al. [29] proposed Multi-Scale Hierarchical Prediction (MSPred), a video prediction framework capable of forecasting video on multiple levels of abstraction and spatio-temporal granularity. By decomposing scenes into objects and modeling their temporal dynamics and relations, sharp multi-step predictions can be learned [30]. Expanding upon the videos, scene flow delves deeper by modeling the 3D+temporal 4D space. As an example, BEVDet4D [31] explicitly maps the result of the previous frame to the current frame through a feature alignment operation, assuming the poses are known. This alignment simplified learning by compensating for ego-motion. Inspired by this, to utilize the historical information, we propagate the features from the previous frame to the current frame via coordinate transformation. Therefore, the model uses the features of the previous frame as a prior to help estimate the current frame. Benefiting from CNN’s property of maintaining spatial order, this method can be used not only for the output layer but also for the latent scene representations.

III. METHOD

3D Semantic Scene Completion aims to jointly estimate both the geometry and semantics of a 3D scene by assigning each voxel a label $L = l_0, l_1, \dots, l_M$, with M semantic classes and l_0 being empty space. To solve this task, SLCF-Net processes sequences of RGB images with associated sparse LiDAR depth, as illustrated in Fig. 2. Our method first computes 2D semantic features and dense depth priors from two input channels, respectively. The 2D semantic features are extracted by an EfficientNet [32] with noisy student training [33]. The relative depth is estimated by the Depth Anything Model and then scaled with raw depth input. Then the 2D semantic features are projected into a 3D voxel grid by the Gaussian-decay Depth-prior Projection (GDP) module. A 3D U-Net learns to generate the complete semantic scene from the 3D features, using a prior propagated from the semantic scene representation of the previous frame compensated for sensor motion.

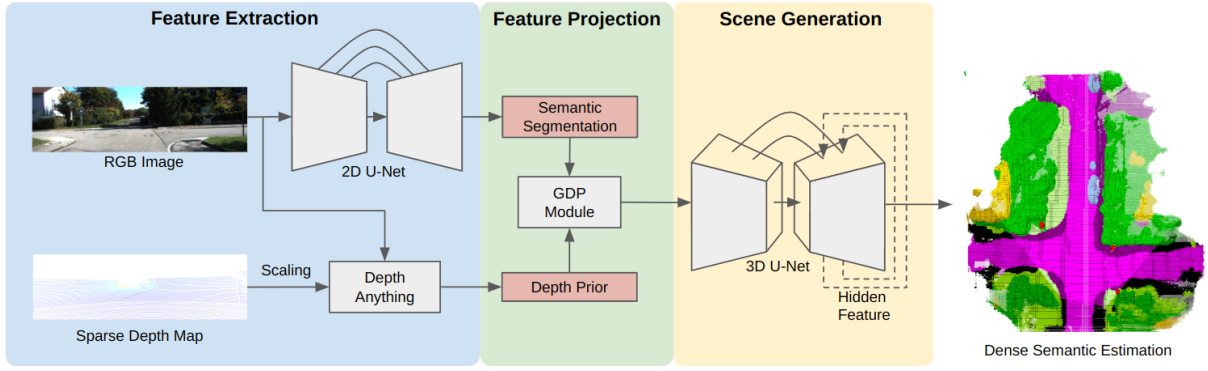


Fig. 2: Overall pipeline of SLCF-Net. Given input sequences consisting of RGB images and a sparse depth map projected from a single sweep point cloud, the process is initiated by extracting the image feature into two channels. The 2D semantic features are extracted by an EfficientNet [32] with noisy student training [33], while the relative depth is estimated by the Depth Anything Model. The relative depth is then scaled based on the sparse depth input to generate depth prior of entire image. Afterward, the Gaussian-decay Depth-prior Projection (GDP) module distributively back-projects the 2D features onto a predefined 3D volume using the depth priors. The 3D features are then fed into a 3D recurrent U-Net, which enables the harness of information from the previous frame. Finally, a dense grid semantic scene is generated as a comprehensive understanding of the environment.

A. 2D-3D Feature Projection

Neural networks have shown prowess in autonomously discerning feature correlations, as exemplified in numerous image-to-image studies. However, the incorporation of extra dimensions in 3D data increases computational requirements, posing a significant challenge to understanding the relationship between 2D and 3D features. Our approach is inspired by the Lift, Splat, Shoot (LSS) mechanism [34]: given known calibration, 2D features can be backprojected into a 3D volume along the line of sight. This inductive bias facilitates learning. Considering a scanning sequence with N frames, each local frame is anchored to the camera coordinate C_i , where $i = 1, \dots, N$. The SemanticKITTI dataset provides the intrinsic calibration and pose estimation of each frame. i.e., we assume the transformation $T_W^{C_i} \in \mathbb{R}^{4 \times 4}$ from world to camera coordinates and the projection $\Pi_i(\cdot)$ to the image plane of C_i to be known. According to the perspective projection model, a 3D point $\mathbf{P} = [X, Y, Z]^T \in \mathbb{R}^3$ can be projected to a pixel $\mathbf{p} = [x, y]^T \in \mathbb{R}^2$ as follows:

$$\tilde{\mathbf{p}} = \Pi_i T_W^{C_i} \tilde{\mathbf{P}}. \quad (1)$$

Here, $\tilde{\mathbf{P}} = [X, Y, Z, 1]^T$ and $\tilde{\mathbf{p}} = [x, y, 1]^T$ represent the homogeneous coordinates of \mathbf{P} and \mathbf{p} , respectively. With Eq. 1, all the voxels in the field of view can be associated with the corresponding 2D feature.

To back-project a pixel to a 3D point, the depth d is necessary. In the SLCF-Net, the depth prior \hat{d} is derived from the relative depth estimation of Depth Anything and scaled according to the LiDAR measurement. Thus, the back-projection is as follows:

$$\hat{\mathbf{P}} = T_{C_i}^W \hat{d} \Pi_i^{-1} \tilde{\mathbf{p}}. \quad (2)$$

For any given pixel coordinate \mathbf{p} , the model will estimate a 3D point \mathbf{P} based on the depth prior \hat{d} . It is worth noticing that by construction the real 3D point is on the line of sight, but the depth prior \hat{d} is based on sensor noise and inference inaccuracies and is hence only a plausible estimate. To model

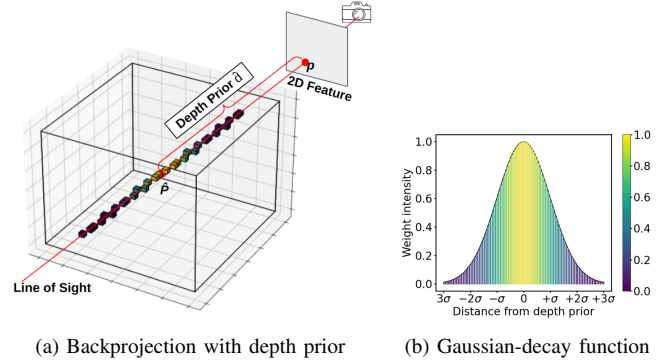


Fig. 3: Gaussian-decay Depth-prior Projection (GDP). (a) a 2D feature located at pixel coordinate \mathbf{p} is projected to voxels in the 3D volume, following the line of sight. Using the depth prior \hat{d} , \mathbf{P} is considered the most probable point and serves as the center of the Gaussian-decay function; (b) Gaussian-decay function weight.

the uncertainty of the depth prior, we employ the Gaussian-decay function:

$$f(\mathbf{P}|\hat{\mathbf{P}}, \sigma) = e^{-\frac{\|\mathbf{P}-\hat{\mathbf{P}}\|^2}{2\sigma^2}}. \quad (3)$$

Here, \mathbf{P} is a voxel passed by the line of sight. $\hat{\mathbf{P}}$ is the estimated point from depth prior \hat{d} , which is also the center of the Gaussian-decay function. As illustrated in Fig. 3, the 2D feature is projected onto all voxels along the line of sight with Gaussian weight. This weight peaks at $\hat{\mathbf{P}}$ and decays with increasing distance from $\hat{\mathbf{P}}$.

In the Gaussian-decay function, the parameter σ signifies our confidence in the depth prior. When there is a significant discrepancy in the depth prior, a larger σ enables the model to jump out from local optima. On the other hand, when the depth prior is accurate, a smaller σ should be adopted to expedite the convergence of the 2D features to the real voxel. Since the depth information is derived from the depth-conditioned Depth Anything pipeline, the choice of σ mainly depends on its noise distribution. The efficacy of the model can be enhanced with an optimal selection of σ , a factor that we will delve into in the ablation studies.

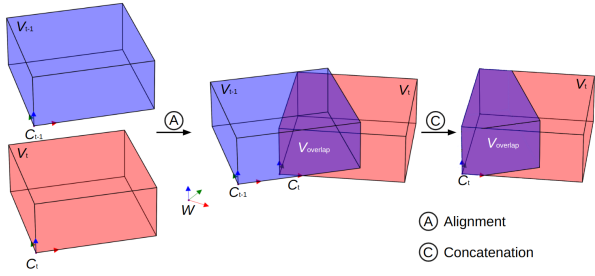


Fig. 4: Concept of temporal feature propagation. In consecutive frames C_{t-1} and C_t , the blue and red cubes represent the defined volumes, respectively. The previous volume V_{t-1} is aligned to the current volume V_t via coordinate transformation. The overlap area, denoted as $V_{overlap}$ and visualized in purple, is repeatedly estimated by neighboring frames so should be consistent. Then the feature located at the same global position is concatenated to propagate information across frames.

B. Temporal Feature Propagation

A few classic frameworks, such as RNN, LSTM, and GRU, have been proposed for sequence learning and widely used in machine translation. RNNs were foundational for sequence learning, and the LSTM and GRU evolved to address gradient instability, letting networks manage longer sequences. To design a framework for temporal feature fusion, we analyzed the characteristics of the SSC task as follows. Although LSTM and GRU excel at extracting high-level features from long sequences, in SSC, the estimation is more dependent on current or nearby frames rather than a long history. Therefore, the primary advantage of LSTM is not needed for SSC tasks. Furthermore, considering that the goal of SSC is to estimate the occupancy and semantics of each voxel, the spatial structure and details are crucial. However, the gate mechanism in LSTM not only destroys the features' spatial structure but also drops details. The 3D U-Net, with its ability to maintain the spatial structure and details in the latent space, is well-suited for SSC. We propose a mapping for the 3D U-Net that propagates the detailed 3D semantic scene representation from the previous frame to the next frame while compensating for sensor motion.

With known poses for each frame, we use the coordinate transformation to align features between consecutive frames. Considering the sequential camera coordinates $C_i, i = 1, \dots, t-1, t, \dots, N$, C_t refers the current frame and C_{t-1} refers the previous frame, which is shown in Fig. 4.

Given the poses of two frames $T_W^{C_{t-1}}, T_W^{C_t}$ and the corresponding volume, V_{t-1}, V_t , a 3D point $P_{C_{t-1}}$ in $V_{overlap}$ can be propagated to a corresponding point P_{C_t} in V_t via the transformation:

$$P_{C_t} = T_W^{C_t} \cdot (T_W^{C_{t-1}})^{-1} \cdot P_{C_{t-1}}. \quad (4)$$

Although P_{C_t} and $P_{C_{t-1}}$ are represented in different coordinate systems, they denote the same physical point in space. This principle of correspondence also applies to 3D features. CNNs naturally maintain the spatial structure of features, so the transformation is suitable for multiple hierarchical levels, which establishes one-to-one correspondences between features. Since this method already handles inter-frame

alignment explicitly, it prevents the model from learning complex relationships caused by ego-motion. As a result, the model can leverage the 3D semantic scene representation from the previous frames as a prior to improve the current semantic scene completion.

C. Losses

SLCF-Net leverages multiple loss functions including the standard cross-entropy loss L_{ce} , the losses L_{mono} proposed by MonoScene [18], and the inter-frame consistency loss L_{con} :

$$L = L_{ce} + L_{mono} + L_{con}. \quad (5)$$

In this study, we delve into the inter-frame consistency loss L_{con} which penalizes discrepancies in the estimations within overlapping regions of adjacent frames. Specifically, we generate a pseudo ground truth using the previously estimated probability distribution and then compute the cross-entropy with the current estimated probability distribution. Furthermore, given that the ground truth is derived from real-world scans, areas obscured by occlusions remain unknown. Consequently, we only compute losses within the known regions.

D. Pipeline for Training, Validation, Testing

SLCF-Net's pipeline for Training, Validation, and Testing is similar to RNNs. During training, due to the large memory size of the 3D data, only two consecutive frames are loaded in a single iteration. With the input sequence $\mathbf{x} = (x_1, \dots, x_t, \dots, x_T)$, estimated sequence $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_t, \dots, \hat{y}_T)$, and hidden states $\mathbf{h} = (h_1, \dots, h_t, \dots, h_T)$, the forward process is described as follows:

$$y_t, h_t = f(x_t, h_{t-1}). \quad (6)$$

For the first frame, when $t = 1$, the previous hidden state h_{t-1} is initialized with an initial hidden feature h_{init} . The model then estimates the scene as y_1 , while simultaneously updating the hidden state h_1 for the subsequent frame. After processing the second frame in the same manner, gradients are propagated backward through time using Backpropagation Through Time (BPTT). Given the brevity of the sequence, the initialization of the hidden state becomes vital. Thus, we treat the initial hidden feature h_{init} as a learnable parameter, enabling the model to learn the optimal initial state across the dataset during training. In our ablation study, we will also evaluate other initialization methods, such as zero initialization and random initialization. For both validation and testing, we utilize the trained h_{init} to initialize h_0 . The model then sequentially infers over the entire sequence.

IV. EVALUATION

A. Experiment Setup

1) *Dataset*: We evaluate the proposed SLCF-Net on the SemanticKITTI dataset, a real-world urban driving scenario dataset. For the SSC task, the official benchmark uses a single LiDAR scan occupancy within a predefined volume as its

TABLE I: Quantitative results on SemanticKITTI validation set.

Method	Input	SC																			mIoU	
		IoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole		traffic-sign
LMSCNet [12]	x_{occ}	29.63	18.64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	40.69	4.82	18.24	0.00	9.28	1.33	14.34	0.02	18.49	0.00	0.00	6.62
AICNet [35]	x_{rgb}, x_{depth}	31.38	15.34	0.00	0.00	<u>5.43</u>	0.00	0.00	0.00	0.00	45.68	12.45	21.34	<u>0.08</u>	13.05	3.26	16.54	3.21	29.03	0.07	0.00	8.71
JS3C-Net [13]	x_{rgb}, x_{pts}	<u>39.63</u>	<u>25.88</u>	0.00	0.00	5.07	<u>7.38</u>	<u>0.74</u>	<u>0.31</u>	0.00	<u>53.01</u>	14.13	<u>27.30</u>	0.07	<u>18.04</u>	<u>4.73</u>	<u>19.92</u>	<u>4.55</u>	<u>29.55</u>	<u>4.34</u>	<u>1.52</u>	<u>11.34</u>
SLCF-Net (ours)	x_{rgb}, x_{depth}	43.64	31.87	0.00	0.15	7.48	8.07	0.84	0.34	0.00	57.50	<u>13.91</u>	29.65	0.45	24.39	12.12	27.66	11.68	36.30	11.66	4.76	14.68

IoU for Scene Completion (SC) task and individual class mIoU for Semantic Scene Completion (SSC) task. **Best** and second best results are highlighted.

input. Specifically, the scene is voxelized into a $256 \times 256 \times 32$ grid with 0.2 m voxels and labeled across 21 categories (19 semantic classes, 1 empty space, and 1 unknown area). The SSC dataset comprises 22 sequences: Sequences 0-7, 9-10 are for training, Sequence 8 for validation, and Sequences 11-21 for testing. The ground truth for the test set is not publicly released. Instead of that, the benchmark provides an online evaluation interface for the generated scenes on the test set but restricts the number of submissions.

Unlike the standard LiDAR occupancy input, our approach utilizes an RGB image combined with a corresponding sparse depth map. The field-of-view for this setup only covers a portion of the entire volume. To ensure a fair comparison with other SSC methods, we retrained the baseline models with the partial data in the field-of-view and also evaluated them only in the field-of-view. Due to the special assessment approach, we evaluate our approach on the original validation set, i.e. we split the dataset as follows. Sequences 0-7 are for training, Sequences 9-10 are for validation, and Sequence 8 for testing. Therefore, we did not use the hidden test set. Moreover, extensive ablation studies and visualization were conducted solely on the released validation set, Sequence 8.

2) *Training Setup*: We trained our model on an NVIDIA RTX A6000 card with 48 GB of memory. Due to the substantial memory demands of 3D data, we adopted a sequential training approach, loading only two consecutive frames of data in a single iteration. This implies that the model only utilized the information from one previous frame. The training spanned 30 epochs utilizing the AdamW optimizer with a weight decay of $1e-4$. The initial learning rate was set at $1e-4$, which was reduced by a factor of 10 at the 15th epoch.

3) *Metrics*: Following the benchmark, the task of scene completion is assessed using Intersection-over-Union (IoU). This metric primarily classifies a voxel as either occupied or empty. Additionally, mean IoU (mIoU) is employed for the task of semantic scene completion across 19 classes. We also apply these metrics to evaluate inter-frame consistency. Specifically, we take the estimations of the neighboring frame and calculate the IoU and mIoU in the overlap area.

B. Performance

Tab. I reports the performance of SLCF-Net and other SSC baselines on the SemanticKITTI validation set. For a fair comparison, the baseline method follows the official

implementation but is retrained with the partial data in the field of view, which is also the area we evaluate all methods. Our method outperforms all baselines in both SC and SSC metrics. More specifically, the SLCF-Net has the highest or second-highest accuracy across all individual classes. Qualitative results for some frames are shown in Fig. 5.

C. Ablation Studies

To gain a deeper understanding of the influence of key parameters and processing steps methods in our method, we conducted a series of experiments. When investigating the impact of a specific factor, we kept other network structures and parameters consistent. We will discuss the findings in the following three aspects.

1) *Variance in GDP*: In the GDP model, the depth prior \hat{d} is inferred by the network, making it not completely reliable and potentially impacting feature mapping. To address this issue, we introduced a Gaussian-decay function, enabling the model to update the depth prior during training. Among its parameters, the variance σ is particularly influential on model performance and the learning process.

Given that in SemanticKITTI the volume is defined as $256 \times 256 \times 32$, we set values for σ as 2^n , where n ranges from 1 to 8. Specifically, when $n = 0$, σ equals one voxel size, suggesting the model has high trust in the depth prior and is less likely to change it during training. Conversely, at $n = 8$, σ equals 256 voxel sizes, covering the entire volume. This implies the model distributes nearly equal weights to all voxels in the line of sight, regardless of the depth prior. Fig. 6 details the results of these experiments.

From Fig. 6a, the model reaches its best performance at $\sigma = 2^4$, indicating this value is the most suitable given the capability of the depth estimation module. As σ increases beyond this, model performance starts to decline. This reveals that giving too little importance to depth priors can degrade the model's estimation ability. When σ is less than 2^4 , the model performance significantly degrades, which signifies that errors in the depth prior can mislead the model into local optima. Further observations from Fig. 6b show that with the increase of σ , the epochs required for optimal performance also increase, signifying that while a larger σ may not drastically reduce performance, it can slow down the learning process.

2) *Initial State in Hidden Layer*: In traditional RNN or LSTM architectures, both zero-initialization and random-

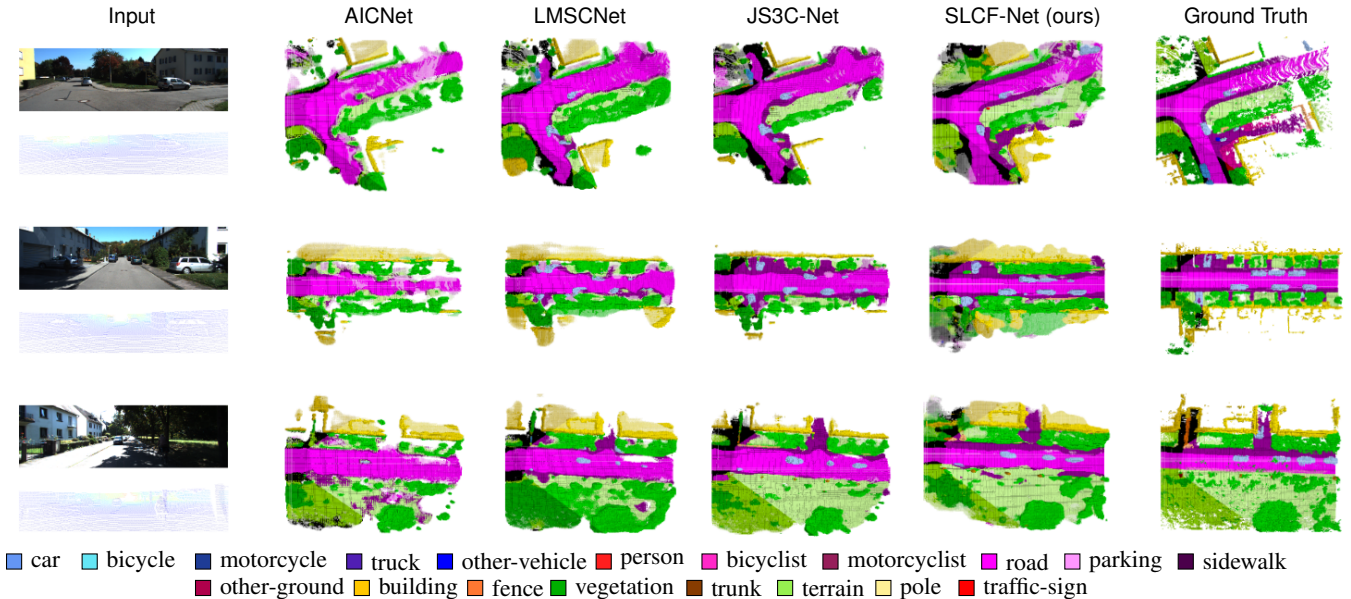


Fig. 5: Qualitative results on SemanticKITTI. From the bird’s eye view, the 19 classes are shown without empty space. The estimated voxels that are located at the unknown region are visualized with 20% opacity. The region located outside of the FoV is shaded.

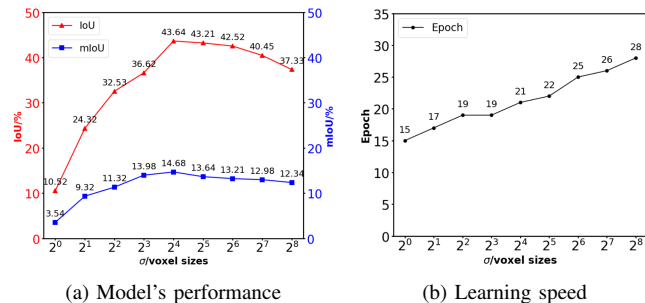


Fig. 6: Influence of σ on model performance and learning process.

initialization of hidden features typically wouldn’t severely impact model performance, as the model can progressively learn and adjust over long sequence data. However, in the context of our work, several factors limit the length of the training sequence, making the initialization strategy profoundly impactful on model performance. We also assessed SLCF-Net’s performance under zero-initialization and random-initialization policies, the results of which are reported in Tab. II. Compared to learnable initialization updated during training, both methods degrade performance.

3) *Trade-off between Accuracy and Consistency*: Using historical data can boost the model’s performance, but adding a penalty for inter-frame inconsistency during training is a double-edged sword. In Tab. III, we evaluate SLCF-Net’s accuracy and consistency with and without L_{con} , comparing it to baselines. Here, consistency is measured using IoU and mIoU in the overlap area of consecutive frames. The results show that, given the incorporation of historical data, SLCF-Net outperforms the baseline in terms of consistency. With

TABLE II: Evaluation of different initialization policies.

Method	IoU	mIoU
zero initialization	35.32	10.34
random initialization	32.64	9.32
learnable initialization (ours)	43.64	14.68

TABLE III: Evaluation of accuracy and consistency.

Method	Accuracy		Consistency	
	IoU	mIoU	IoU	mIoU
LMSCNet [12]	30.04	6.70	26.35	5.34
JS3C-Net [13]	43.88	11.34	31.54	6.75
AICNet [35]	31.38	8.71	20.35	4.32
SLCF-Net (ours)	39.64	10.63	35.62	8.63
SLCF-Net w/o L_{con}	43.64	14.68	33.65	7.62

the addition of L_{con} , consistency further improves, but at the cost of reduced accuracy. This trade-off arises because when there’s a discrepancy between the earlier estimation and the current truth, the model has to align both. While L_{con} smoothens sequence estimation, accuracies in one frame can adversely influence the subsequent frame’s results.

V. CONCLUSIONS

For the task of semantic scene completion (SSC), we proposed SLCF-Net which leverages sequences of RGB images and sparse LiDAR depth maps as input. Central to SLCF-Net is its feature fusion mechanism, which integrates a 2D-to-3D feature projection and an inter-frame feature propagation. We tested SLCF-Net using the SemanticKITTI dataset and compared its performance with other SSC techniques.

To further improve SLCF-Net’s efficacy, we conducted a series of targeted experiments. These explored the effects of the Gaussian variance, the strategy of initializing hidden features, and the implementation of an inter-frame consistency loss. Although SLCF-Net demonstrates notable advantages, it presents a trade-off between accuracy and consistency, which means effectively using historical information is still an open challenge. Additionally, the current method is designed for static environments, thereby oversimplifying the dynamic objects in real-world urban contexts. Propagating semantic scene representations using estimated scene flow [36], [37] is a promising direction for future research.

REFERENCES

- [1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.
- [2] P. Sun, H. Kretschmar, X. Dotiwala, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo Open Dataset," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2446–2454.
- [3] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [4] G. Fahim, K. Amin, and S. Zarif, "Single-view 3D reconstruction: A survey of deep learning methods," *Computers & Graphics*, vol. 94, pp. 164–190, 2021.
- [5] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," *arXiv preprint arXiv:2401.10891*, 2024.
- [6] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *4th Eurographics Symposium on Geometry Processing*, vol. 7, 2006.
- [7] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Transactions on Graphics*, vol. 32, no. 3, pp. 1–13, 2013.
- [8] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1746–1754.
- [9] L. Roldao, R. De Charette, and A. Verroust-Blondet, "3D semantic scene completion: A survey," *International Journal of Computer Vision (IJCV)*, vol. 130, no. 8, pp. 1978–2005, 2022.
- [10] J. Li, Y. Liu, X. Yuan, C. Zhao, R. Siegwart, I. Reid, and C. Cadena, "Depth based semantic scene completion with position importance aware loss," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 1, pp. 219–226, 2019.
- [11] S.-C. Wu, K. Tateno, N. Navab, and F. Tombari, "SCFusion: Real-time incremental scene reconstruction with semantic completion," in *International Conference on 3D Vision (3DV)*, 2020, pp. 801–810.
- [12] L. Roldao, R. de Charette, and A. Verroust-Blondet, "LMSCNet: Lightweight multiscale 3D semantic completion," in *International Conference on 3D Vision (3DV)*, 2020, pp. 111–119.
- [13] X. Yan, J. Gao, J. Li, R. Zhang, Z. Li, R. Huang, and S. Cui, "Sparse single sweep LiDAR point cloud segmentation via learning contextual shape priors from scene completion," in *National Conference on Artificial Intelligence (AAAI)*, vol. 35, no. 4, 2021, pp. 3101–3109.
- [14] R. Cheng, C. Agia, Y. Ren, X. Li, and L. Bingbing, "S3CNet: A sparse semantic scene completion network for LiDAR point clouds," in *Conference on Robot Learning (CoRL)*, 2021, pp. 2148–2161.
- [15] M. Zhong and G. Zeng, "Semantic point completion network for 3D semantic scene completion," in *Europ. Conference on Artificial Intelligence (ECAI)*. IOS Press, 2020, pp. 2824–2831.
- [16] Y. Cai, X. Chen, C. Zhang, K.-Y. Lin, X. Wang, and H. Li, "Semantic scene completion via integrating instances and scene in-the-loop," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 324–333.
- [17] I. Cherabier, J. L. Schonberger, M. R. Oswald, M. Pollefeys, and A. Geiger, "Learning priors for semantic 3D reconstruction," in *Europ. Conference on Computer Vision (ECCV)*, 2018, pp. 314–330.
- [18] A.-Q. Cao and R. de Charette, "MonoScene: Monocular 3D semantic scene completion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 3991–4001.
- [19] G. Riegler, A. O. Ulusoy, H. Bischof, and A. Geiger, "OctNetFusion: Learning depth fusion from data," in *International Conference on 3D Vision (3DV)*, 2017, pp. 57–66.
- [20] J. Czarnowski, T. Laidlow, R. Clark, and A. J. Davison, "DeepFactors: Real-time probabilistic dense monocular SLAM," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 2, pp. 721–728, 2020.
- [21] S. Bultmann and S. Behnke, "3D semantic scene perception using distributed smart edge sensors," in *International Conference on Intelligent Autonomous Systems (IAS)*, 2022, pp. 313–329.
- [22] S. Bultmann, R. Memmesheimer, and S. Behnke, "External camera-based mobile robot pose estimation for collaborative perception with smart edge sensors," *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [23] J. Hau, S. Bultmann, and S. Behnke, "Object-level 3D semantic mapping using a network of smart edge sensors," in *IEEE International Conference on Robotic Computing (IRC)*, 2022, pp. 198–206.
- [24] S. Bultmann, J. Quenzel, and S. Behnke, "Real-time multi-modal semantic fusion on unmanned aerial vehicles with label propagation for cross-domain adaptation," *Robotics and Autonomous Systems*, vol. 159, p. 104286, 2023.
- [25] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, no. 64–67, p. 2, 2001.
- [26] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [28] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27, 2014.
- [29] A. Villar-Corrales, A. Karapetyan, A. Boltres, and S. Behnke, "MSPred: Video prediction at multiple spatio-temporal scales with hierarchical recurrent networks," *British Machine Vision Conference (BMVC)*, 2022.
- [30] A. Villar-Corrales, I. Wahdan, and S. Behnke, "Object-centric video prediction via decoupling of object dynamics and interactions," in *IEEE International Conference on Image Processing (ICIP)*, 2023, pp. 570–574.
- [31] J. Huang and G. Huang, "BEVDet4D: Exploit temporal cues in multi-camera 3D object detection," *arXiv preprint arXiv:2203.17054*, 2022.
- [32] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.
- [33] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 687–10 698.
- [34] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D," in *16th European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 194–210.
- [35] J. Li, K. Han, P. Wang, Y. Liu, and X. Yuan, "Anisotropic convolutional networks for 3D semantic scene completion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3351–3359.
- [36] P. Schütt, R. A. Rosu, and S. Behnke, "Abstract flow for temporal semantic segmentation on the permutohedral lattice," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5139–5145.
- [37] Y. Lin and H. Caesar, "ICP-Flow: LiDAR scene flow estimation with ICP," *arXiv preprint arXiv:2402.17351*, 2024.