SIUNet: Sparsity Invariant U-Net for Edge-Aware Depth Completion

Avinash Nittur Ramesh Fabio Giovanneschi María A. González-Huici
Fraunhofer FHR
Wachtberg, Germany
avinash.ramesh@fhr.fraunhofer.de

Abstract

Depth completion is the task of generating dense depth images from sparse depth measurements, e.g.*, LiDARs. Existing unguided approaches fail to recover dense depth images with sharp object boundaries due to depth bleeding, especially from extremely sparse measurements. State-of-the-art guided approaches require additional processing for spatial and temporal alignment of multi-modal inputs, and sophisticated architectures for data fusion, making them non-trivial for customized sensor setup. To address these limitations, we propose an unguided approach based on U-Net that is invariant to sparsity of inputs. Boundary consistency in reconstruction is explicitly enforced through auxiliary learning on a synthetic dataset with dense depth and depth contour images as targets, followed by fine-tuning on a real-world dataset. With our network architecture and simple implementation approach, we achieve competitive results among unguided approaches on KITTI benchmark and show that the reconstructed image has sharp boundaries and is robust even towards extremely sparse LiDAR measurements.*

1. Introduction

Depth completion is a popular topic in the automobile industry and robotics community, especially in the field of SLAM [29], object detection [3, 15, 42], segmentation [28, 50, 52], etc. It aims at recovering missing depth values in sparse measurements. Modern sensors such as IR depth cameras and stereo camera modules directly provide dense depth images. However, they are not suitable for long range applications, and their performance significantly deteriorates in low-light conditions making them unsuitable for outdoor environment. Conventional LiDARs, on the other hand, excel in this area with capability to scan the environment up to around 200m, although with limited vertical and horizontal resolution due to mechanical constraints. With advances in sensor technology, coherent MEMs-based LiDAR systems [6, 19, 48] provide better distance resolution, faster scan rates, and allow fine control over vertical and horizontal resolution by electronically steering a mirror. Cwalina *et al*. [6], proposed a high-resolution MEMs based LIDAR sensor able to achieve faster scan rates by performing very sparse random scanning of the scenario, which is particularly convenient in highly dynamic scenarios. They relied on post-processing methods based on compressive sensing and deep learning for successfully recovering dense depth images. LiDAR sensors typically provide sparse point clouds compared to the depth camera images, and require additional processing (3D to 2D projection and depth completion) to facilitate higher-level tasks, such as obstacle avoidance, object recognition, segmentation, etc. Developing a generalized solution for depth completion is challenging because of sparsity in the input, irregular spacing between the scanned points, and the usage of different LiDARs systems depending on the application requirement. A naive way of obtaining dense images is through simple interpolation. But, this results in depth bleeding or blurring around depth discontinuities, especially for extremely sparse measurements. State-of-the-art (SoTA) deep learning approaches complement sparse LiDAR measurements

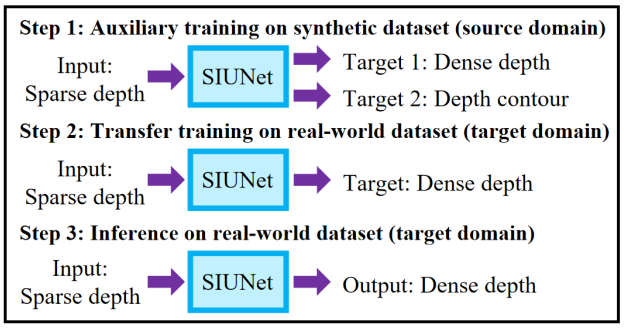


Figure 1: SIUNet is trained in two stages, *i.e*., auxiliary learning followed by transfer learning. Depth contour images are generated from synthetic dataset and used as target during training to enforce boundary consistency in the dense depth output. Sparse depth is used as the only input during training and inference

with additional data obtained from other sensors, such as stereo camera images, RGB images, etc., to obtain dense depth images [17, 24, 47, 53]. However, these methods are non-trivial [14, 33], because they require sophisticated neural network architectures to perform data fusion, handle noisy data from different sensors, and address the problem of occlusion resulting from small shift in the placement of sensors. Several unguided methods [5, 10, 22, 41] exist that rely only on the availability of sparse LiDAR measurements. The drawback of these methods are that they fail to recover sharp object boundaries due to the sparsity of inputs, and unavailability of RGB images that can provide semantic cues.

In this work, we address some of the above mentioned shortcomings of guided and unguided depth completion approaches. To this end, we propose a simple yet effective sparsity-invariant convolutional neural network based on U-Net [35] architecture for producing dense images with sharp object boundaries. Our network relies on the availability of only LiDAR measurements in the target domain as shown in Fig. 1, and is invariant to the sparsity of input. We followed multi-task feature learning [2, 11, 25], where we initially trained our network on a synthetic dataset with sparse depth image as input, and dense depth image along with a corresponding depth contour image (generated from semantic segmentation images of source domain) as targets. This was followed by transfer learning on a real-world dataset (target domain) where only LiDAR measurements are used as input and target.

We summarize the key contributions of our work here:

- We propose a sparsity invariant U-net architecture that takes sparse depth image as the only input, and produces a dense depth image and an auxiliary/ vestigial depth contour image. The speciality of our approach is that there are no branch-outs [50] in our network, resulting in a very simple architecture with end-to-end feature sharing.
- We propose a novel method that generates target depth contours and allows the network to explicitly learn structural information from them by employing a novel loss function.
- We show the competitiveness of our approach by comparing qualitatively and quantitatively with existing unguided approaches and finally, demonstrate the robustness of our network towards extremely sparse LiDAR measurements.

2. Related work

2.1. Guided depth completion

Guided depth completion methods [17, 24, 47, 53] outperform traditional hand-crafted methods and unguided methods for depth completion by a wide margin. These methods require additional data along with sparse LiDAR measurements as input, to guide the network. These inputs are fused through early-fusion, hybrid-fusion or late-fusion techniques [28, 37, 46, 53]. Since, RGB images provide strong cues on semantic and contextual information, current state-of-the-art solutions [14, 17, 24, 47, 53] rely on image guidance and attention-based techniques. Recovery of object boundaries is crucial to the task of depth completion, because it facilitates distinguishing objects in a scene. So, by fusing multiple inputs, these methods perform exceedingly well in attaining boundary consistency in the dense depth outputs. However, these methods are tedious and non-trivial [14, 21, 33, 54] especially for customized end-user systems, because they require sophisticated neural network architectures to perform data fusion, handle noisy sensor data, remove outliers due to occlusion as a result of small displacement in the viewpoint of sensors, and handle motion artifacts in dynamic scenarios due to different acquisition time, etc. The dependency of these methods on additional sensors results in an overall increase in the cost of end-user systems, such as autonomous vehicles, robots, etc.

2.2. Unguided depth completion

Unguided depth completion methods [1, 5, 10, 22, 23, 30, 41] rely on the availability of only sparse measurements as input. This greatly enhances simplicity of the overall system because additional tasks such as, data fusion, removal of outliers, and sensor synchronization are not required. Early approaches [1, 22, 30] produced dense depth images based on traditional hand-crafted rules or image processing techniques. These methods are based on prior knowledge and are not robust to variability and uncertainty in sparse measurements. A naive way to produce dense depth output in a learning-based approach would be to train a convolutional regression network to directly map sparse LiDAR input to dense depth output. Li *et al*. [23] assigned zeros to missing depth points in the sparse input and chose to train with a standard CNN. Uhrig *et al*. [41] demonstrated that such approaches produce sub-optimal results because they are sensitive to variation in sparse measurements. To address this issue, they proposed sparsity invariant CNN, which explicitly considers the location of missing points during convolution. Jaritz *et al*. [28] proposed that a standard CNN architecture can perform well with ad-hoc training process, *i.e*., by introducing varying levels of sparsity in the input during training. However, due to the unavailability of RGB images or semantic cues in the inputs, unguided methods tend to produce smooth edges due to depth bleeding around object boundaries. To address this problem, several loss functions have been previously designed [26, 43, 40, 44, 49]. Nevertheless, dense depth images produced by unguided depth completion tech-

niques produce inconsistent boundaries [25, 47], especially for highly sparse inputs.

2.3. Multi-task feature learning

Multi-task feature learning [11] enforces a network to learn independent yet related tasks simultaneously, resulting in performance improvement in both the tasks. The idea is to exploit commonalities between the tasks and transfer knowledge across tasks implicitly by using shared features. Jaritz *et al*. [28] proposed an encoder-decoder based on NASNet [55] and used the same network with slight adaptation to achieve both depth completion and semantic segmentation. Ye *et al*. [50] initially trained two networks independently for the purpose of depth completion and semantic segmentation. Then they proposed an encoder-decoder based network which shares the features of both these networks to jointly estimate dense depth image and semantically segmented image in a knowledge amalgamation setup. Lu *et al*. [25], on the other hand, employed auxiliary learning [34] approach, which is a variant of multi-task feature learning. Their main focus was on generating dense depth output. But they simultaneously also produce gray-scale image as an auxiliary task to assist in depth completion.

Our work also employs auxiliary learning technique, motivated by the work of Lu *et al*. [25]. In contrast to their work, our network generates dense depth image and depth contour image. The speciality of our proposed approach is that there is no branching in the network, resulting in a simple architecture. The reason is that both primary and auxiliary tasks in our case are in the same domain, i.e., depth. This results in end-to-end sharing of features, enhancing the primary task even further. Additionally, their work relies on RGB images of target domain as auxiliary target during the training process. However, our solution solely relies on the availability of LiDAR data, without the need for target domain semantic priors during training, allowing the end-user system to rely solely on a LiDAR sensor.

3. Methodology

3.1. Problem formulation

We presume that the end-user system/ sensor setup provides only LiDAR measurements, and does not provide data from any other sensor modalities, such as RGB camera. We consider that the sparsity and scan patterns in LiDAR measurements can vary significantly. With these assumptions, our goal is to obtain dense depth image with sharp object boundaries by relying only on sparse LiDAR measurements as input to the network. We propose a simple sparsity invariant neural network based on U-Net [35] architecture. We train this network through auxiliary learning approach on synthetic dataset SYNTHIA [36] (source domain), followed by transfer learning on real-world dataset KITTI [13] (target domain). In the auxiliary learning approach, our network is trained to optimize over two targets, i.e., dense depth and depth contour images by employing a novel loss function. Generation of synthetic LiDAR data and depth contour ground-truth are discussed in Sections 3.3 and 3.4. We continue the training by performing transfer learning with real-world LiDAR measurements as input and semi-dense ground-truth of KITTI as target. A summary of our approach is shown in Fig. 1.

3.2. Network architecture

Our network architecture is shown in Figure 2b, which is a fully sparse convolutional neural network based on U-Net architecture. It has an encoder consisting of multiple down-sampling stages, followed by a decoder consisting of multiple up-sampling stages, with residual connection between each stage. Low-level structural features are extracted at shallow stages and high-level contextual features are captured at deeper stages. The residual connections allow fusion of multi-resolution features at different hierarchical levels of encoder and decoder. At the bottleneck, all the feature maps are with $\frac{1}{16}$ resolution of the input. We employed and adapted sparse convolution layers proposed by Uhrig *et al*. [41], to perform convolution, transpose convolution, and feature concatenation in our network. A binary mask indicating the validity of input pixels is propagated through the network along with the features. Down-sampling is performed by strided sparse convolutional layers, and up-sampling is performed by strided sparse transpose convolutional layers. Features are concatenated as suggested by Huang *et al*. [18]. Due to the use of sparse convolutional layers for upsampling, we mitigate depth bleeding effect at object boundaries as demonstrated in Section 4.4. Our network produces two outputs at the last layer, but does not require branch-outs [50] because both outputs are in the same domain, *i.e*., depth.

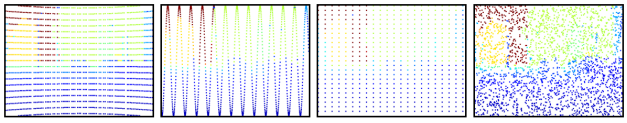
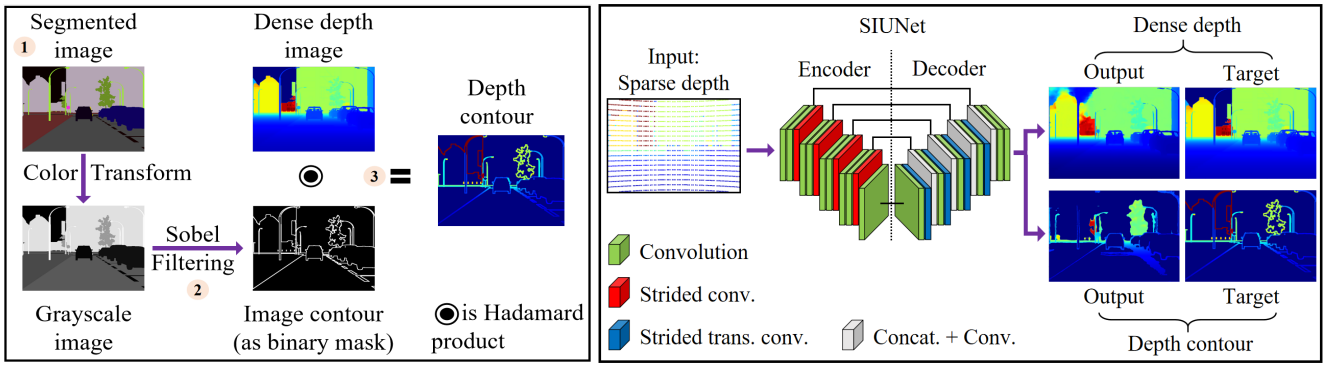


Figure 3: Randomly and structurally sparsified depth ground-truth of SYNTHIA. The images have been dilated for visibility (best viewed in color)

3.3. Training details

SYNTHIA [36] dataset provides dense depth images that can be used as target, but it does not provide LiDAR data for the input. So, dense depth images were randomly or structurally sparsified by annihilating some pixels using Bernoulli sampling, and subsequently used as input. Examples of a few patterned depth images are shown in Fig. 3.



(a) Generation of target depth contour

(b) Network architecture of SIUNet

Figure 2: SIUNet in auxiliary learning setup (source domain): Initially, target depth contour images are generated by exploiting semantic segmentation images of SYNTHIA [36] dataset. Then the network is trained on SYNTHIA dataset with sparsified depth images as the only input, and dense depth and depth contour images as targets (best viewed in color)

Structural LiDAR-like patterns were created by projecting depth pixels from camera image plane to 3D spherical coordinate system, using intrinsic camera calibration parameters, and then filtering out the point cloud data depending on the required azimuth and elevation resolutions. Several patterns like these were created and used as input for our network during training. On-the-go data augmentation techniques such as adding zero-mean gaussian noise with depth-dependent variance, uniform random noise, intensity based drop-off [7], random cropping, and horizontal flipping were performed on both KITTI and SYNTHIA datasets. The idea was to obtain a robust network that is invariant to input patterns or sparsity.

3.4. Depth contour generation

SYNTHIA dataset provides semantically segmented images with corresponding dense depth images. Semantically segmented images are converted to grayscale color space, and edges are extracted from them by performing convolution with a sobel filter. The result of this operation is used as a binary mask over dense depth images to finally obtain depth contour images, as shown in Fig. 2a. The idea of using depth contour instead of RGB or segmented image is that the domain of depth contour and dense depth image is the same, *i.e.*, depth. This greatly simplifies the network architecture by reducing the number of trainable parameters compared to the architectures with branch-outs. Additionally, depth contour images enforce boundary consistency because they provide structural cues in lieu of semantic cues, making the network edge-aware and generic towards unseen semantic classes in the datasets. This has been shown in Section 4.5, where our network is evaluated on an indoor dataset inspite of training on outdoor dataset.

3.5. Loss function

Given a sparse depth input, x, we intend to obtain a dense depth output, $\tilde{x}_d$, by optimizing the network parameters, θ. Optimization is done by minimizing a loss function, $\mathcal{L}$, that calculates error between reconstructed outputs, $\tilde{x}_d$ and $\tilde{x}_c$, and target ground-truth, x_d^* and x_c^*, as shown in Eq. (1). θ^* indicates optimal network weights that is obtained after the training process. In the auxiliary learning process, we have two targets and employ two loss functions. First one is for the primary task of depth completion, shown in Eqs. (2) and (3), where dense depth image, x_d^*, is the target. Here, a depth mask, m, is used to indicate valid depth pixels in target. The second loss function is for the auxiliary task of depth contour generation, shown in Eqs. (4) and (5). Semantically segmented image, s, is used to generate target depth contour image, $x_c^* = x_d^* \circ c$. Here, a contour mask, c, is used to indicate valid contour pixels in the target depth contour. Finally, for the transfer learning task, we only use Eqs. (2) and (3). The depth and contour masks are used for both MAE and RMSE calculations.

$$\theta^* = \arg\min_{\theta} \mathcal{L}(x_d^*, x_c^*, \tilde{x}_d, \tilde{x}_c) \tag{1}$$

$$m_{u,v} = \begin{cases} 1, & x^*_{d_{u,v}} \neq 0 \\ 0, & x^*_{d_{u,v}} = 0 \end{cases} \tag{2}$$

$$\mathcal{L}^D_{\text{MAE}} = \frac{\sum_{i,j} \left| x^*_{d_{i,j}} - \left(\tilde{x}_{d_{i,j}} \circ m_{i,j}\right) \right|}{\sum_{i,j} m_{i,j}} \tag{3}$$

$$c_{u,v} = \begin{cases} 1, & \nabla s_{u,v} \neq 0 \\ 0, & \nabla s_{u,v} = 0 \end{cases} \tag{4}$$

$$\mathcal{L}^C_{\text{MAE}} = \frac{\sum_{i,j} \left| x^*_{c_{i,j}} - \left(\tilde{x}_{c_{i,j}} \circ c_{i,j}\right) \right|}{\sum_{i,j} c_{i,j}} \tag{5}$$

Total loss, $\mathcal{L}_{\text{MAE}}^{Tot}$, of auxiliary learning process, is calculated as a weighted combination of $\mathcal{L}_{\text{MAE}}^{D}$ and $\mathcal{L}_{\text{MAE}}^{C}$, as shown in Eq. (6).

$$\mathcal{L}_{\text{MAE}}^{Tot} = w_d \cdot \mathcal{L}_{\text{MAE}}^{D} + w_c \cdot \mathcal{L}_{\text{MAE}}^{C} \tag{6}$$

4. Experiments and results

Datasets: SYNTHIA [36] is a synthetic dataset that provides corresponding images of RGB, depth, and semantics, of dimensions (640×480), for urban and highway scenarios. A total of $96,348$ corresponding images are provided as training set, and $29,850$ as testing set. We employed SYNTHIA dataset as source domain only for auxiliary learning, and exploited semantic images for generation of depth contours. We employed KITTI [13] dataset as target domain for transfer learning, where training and inference depend only on LiDAR measurements. KITTI dataset provides sparse velodyne LiDAR depth maps, and semi-dense depth ground-truth images which were center-cropped to 1216×352. A total of $85,898$ pairs of these are provided as training set, and $15,920$ pairs as validation set. Performance of our network was evaluated on the test set of KITTI benchmark [13].

Evaluation Metrics: In accordance with the KITTI depth completion benchmark, we evaluated our model with RMSE (mm) and MAE (mm) for depth, and iRMSE (km^{-1}) and iMAE (km^{-1}) for inverse depth. We have sorted the models in decreasing order of MAE metric in Table 1.

Implementation Details: We used PyTorch [32] for the implementation of our networks, and trained them on NVIDIA Quadro RTX 4000 GPU with a batch size of 4. We chose Adam [20] optimizer with a constant learning rate of 0.001, without weight decay, and trained for 20 epochs on SYNTHIA dataset, and 10 epochs on KITTI dataset. We initialized the weights of our network with Kaiming normal distribution [16], and used $w_d = w_c = 1$, for the calculation of loss in auxiliary learning. For transfer learning, the weights of only the last three layers were optimized, and depth contour targets were not used. The idea was to use the learned weights from auxiliary learning task as weight initialization for the transfer learning task.

4.1. Qualitative analysis

We present our qualitative results in Fig. 4, and compare it with the SoTA unguided methods, such as Spade-sD [28], pNCNN [8], Sparse-to-Dense [26], PSM [54]. We consider three examples, and provide closeup views of the region of interest in the bottom right corner of each reconstructed image. It is evident that qualitatively our model outperforms all the methods in terms of boundary consistency and structural correctness. In Examples 1 and 3, we can clearly see depth bleeding occurring at the boundary of the cars for other methods. However, our model produces sharp reconstructions. In Example 2, the unguided approaches tend to produce discontinuities, however, our model is able to recover depth and structure of the traffic sign board. The reason for such good reconstructions can be attributed to auxiliary learning with depth contour images. More evidence for this is provided in Section 4.3.

Table 1: Numerical comparison of our approach with SoTA unguided approaches shows that our approach produces competitive results. The error metrics were calculated on KITTI depth completion benchmark. ↓ indicates smaller is better

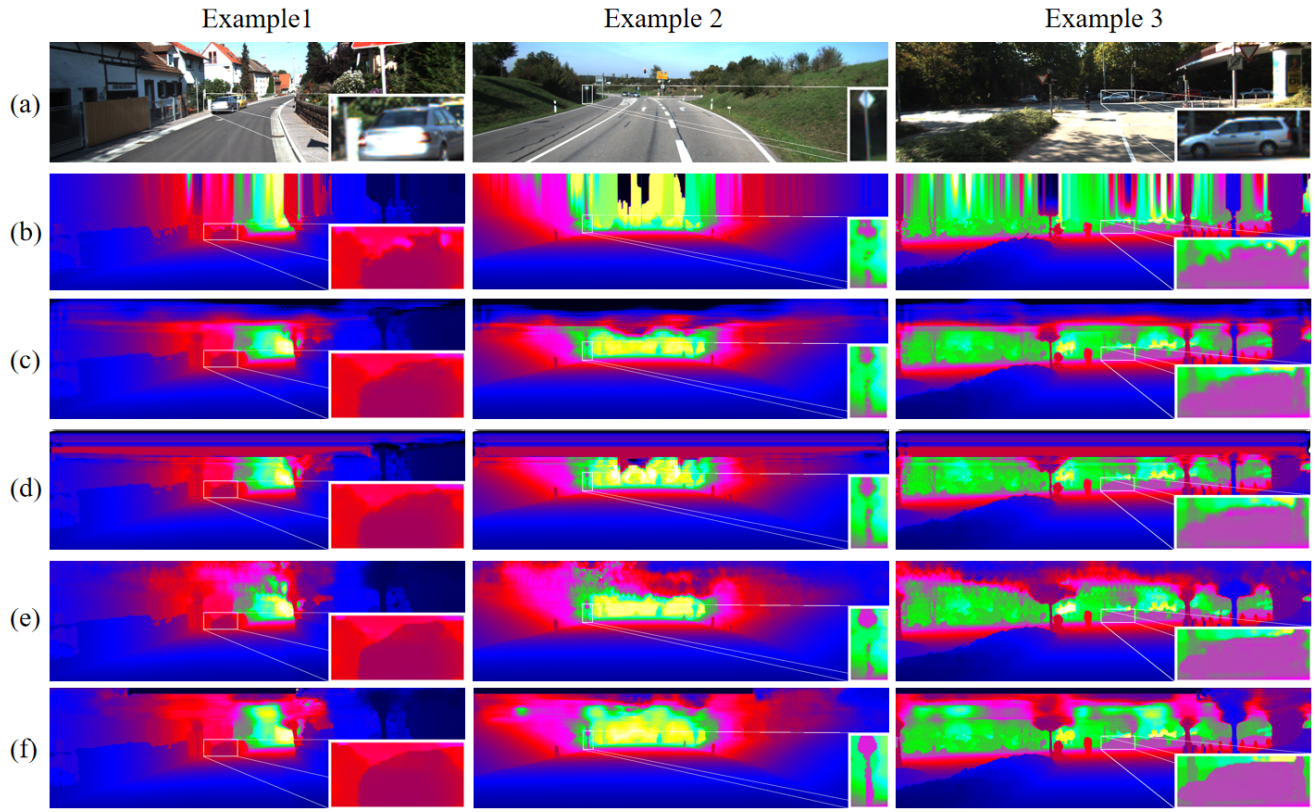
Method	iRMSE ↓ (km^{-1})	iMAE ↓ (km^{-1})	RMSE ↓ (mm)	MAE ↓ (mm)
SICNN [41]	4.94	1.78	1601.33	481.27
ADNN [5]	59.39	3.19	1325.37	439.48
NCNN [9]	4.67	1.52	1268.22	360.28
IP-Basic [22]	3.78	1.29	1288.46	302.60
PSM [54]	3.76	1.21	1239.84	298.30
StoD(d) [26]	3.21	1.35	**954.36**	288.64
pNCNN [8]	3.37	1.05	960.05	251.77
Spade-sD [28]	**2.60**	0.98	1035.29	248.32
SIUNet (Ours)	2.73	**0.96**	1026.61	**227.28**

4.2. Numerical analysis

Table 1 presents quantitative results of our method and SoTA approaches evaluated on KITTI depth completion benchmark. Our model achieves the best **MAE = 227.28**, and **iMAE = 0.96**, ranking first among them. Our RMSE and iRMSE results are also better than most of the other models, demonstrating the effectiveness of our approach. Our model was initially trained on SYNTHIA dataset in an auxiliary learning setup with l_1 loss, and then on KITTI dataset with only semi-dense groundtruth images as targets during transfer learning.

4.3. Ablation study

To show the effectiveness of our approach, we performed various experiments by incrementally adding the building blocks of our framework. We show quantitative results of ablation study in Table 2. We studied the impact of unavailability of dense ground-truth, and the impact of different training methods such as conventional learning, transfer learning (TL), auxiliary learning (AL), and zero-shot learning (ZSL), and their combinations, and finally the impact of the training loss. (Test 1) We started with a naive way of implementing depth completion under the assumption that the ground-truth depth images were not available. In this con-



(a) RGB images for reference, (b) PSM [54], (c) StoD(d) [26], (d) pNCNN [8], (e) Spade-sD [28], (f) SIUNet (Ours)

Figure 4: Qualitative comparison of our approach with SoTA unguided depth completion approaches sorted in decreasing order of MAE from top to bottom. The closeup views of our method show sharpness along object boundaries and structural correctness. Depth bleeding can be observed in the reconstructions of other methods

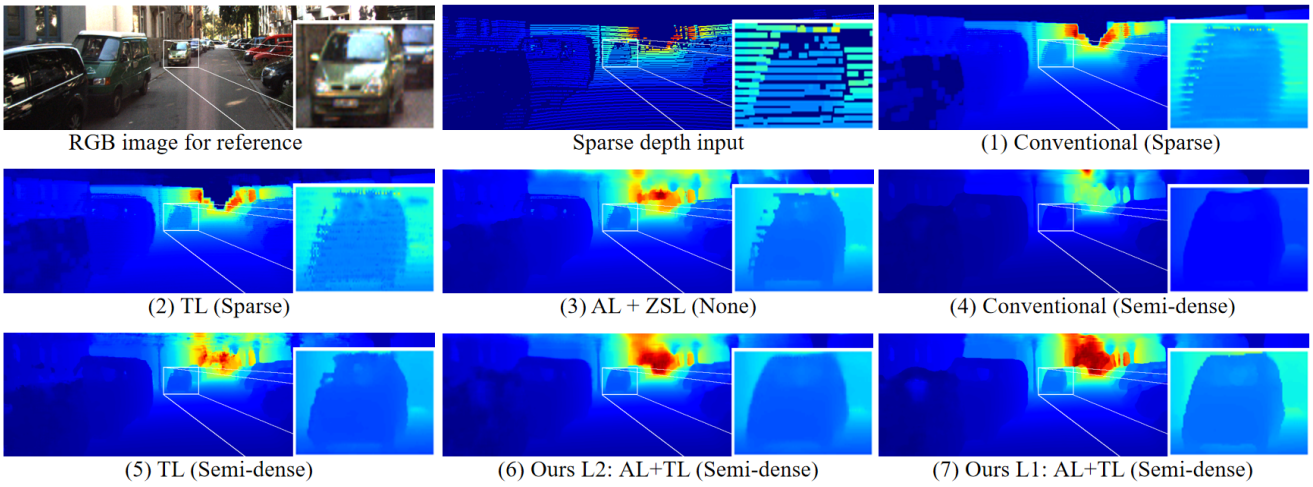


Figure 5: Ablation study with qualitative comparison of different training methods such as conventional, transfer learning (TL), auxiliary learning (AL), and zero-shot learning (ZSL). Before training on KITTI, TL and AL approaches were trained on SYNTHIA dataset. Sparse depth input has been dilated for visibility

Table 2: Ablation study with different training targets, and training methods such as conventional, transfer learning (TL), auxiliary learning (AL), and zero-shot learning (ZSL). The errors were calculated with respect to semi-dense ground-truth images of KITTI depth completion validation set. Before training on KITTI, TL and AL approaches were trained on SYNTHIA dataset

Test	Training method	Training target	MAE (mm)
1	Conventional	Sparse	495
2	TL	Sparse	421
3	AL + ZSL	None	364
4	Conventional	Semi-dense	267
5	TL	Semi-dense	266
6	AL+TL (Ours L2)	Semi-dense	249
7	AL+TL (Ours L1)	Semi-dense	**224**

ventional setting, we trained our model with sparse LiDAR depth images of KITTI itself as input as well as target. We evaluated our model's performance on KITTI semi-dense ground-truth images. It can be seen from the corresponding image in Fig. 5, that this creates line artifacts in the reconstructed output. (Test 2) To improve the reconstruction, we resorted to training our model in a conventional setting on SYNTHIA dataset, and subsequently performed transfer learning on KITTI, but with sparse depth as target. Although SYNTHIA provides dense depth images, it can be seen that after transfer learning, we still find some unwanted artifacts in the reconstruction. This can be attributed to the high sparsity of target image during transfer learning. To justify this, (Test 3) we performed another experiment with auxiliary learning on SYNTHIA dataset, and evaluated it directly on KITTI dataset. The importance of dense depth targets can clearly be visualized at this stage. We propose that for scenarios where real-world ground-truth images and other sensor data are not available, we could opt for the proposed zero-shot learning approach. The speciality of this is its robustness to sparsity and pattern. (Test 4) We further evaluated the model's performance when trained in conventional learning, but with semi-dense depth images of KITTI as targets. This method significantly outperforms the zero-shot learning approach. This is because in zero-shot learning approach, the model is trained in a generic manner, without learning target domain specifics. (Test 5) Further tests, show that denser depth targets indeed improve the quality of reconstruction in terms of structural correctness. (Tests 6, 7) Finally, we show that training models with MAE loss yields sharper object boundaries in comparison to training with RMSE loss. In Fig. 6, we also show the effectiveness of using sparsity invariant convolutions (in SIUNet) instead of regular convolutions (in UNet) for robustness against varying levels of input sparsity.

4.4. Sparsity invariance

We evaluated our model's performance for varying levels of sparsity in the input depth image. To induce sparsity, we randomly annihilated only the valid pixels of an already sparse input depth image using Bernoulli sampling. Figure 6 shows qualitative and quantitative results of reconstructed depth images for varying levels of sparsity (indicated in % on top) induced in the sparse input image. We followed an ad-hoc training approach as proposed by Jaritz *et al*. [28], where we varied the sparsity of input randomly in between 0% and 99%. It is evident that our model performs well for different sparsity levels, while maintaining sharp object boundaries in the reconstructed images. It is important to note that at 95% sparsity, the input image contains only $\approx 1,000$ valid depth pixels. With such extremely sparse depth image as input to our network, we were able to reconstruct $\approx 2,70,000$ pixels, amounting to a ratio of $\approx 0.37\%$, with an MAE of 512 mm. Compared to the works of Jaritz *et al*. [28], and Huang *et al*. [18], our approach does not smear the reconstruction around depth discontinuities for extremely sparse inputs. We also show the effectiveness of using sparsity invariant convolutions (in SIUNet) instead of regular convolutions (in UNet) for robustness against varying levels of input sparsity. Here, both UNet and SIUNet were trained under the same conditions, and as illustrated in Fig. 1.

4.5. Generalization ability

We also performed experiments on NYU v2 indoor dataset [39], and numerically compared our results with guided methods as shown in Table 3. Our model provides competitive results on the target domain (indoor) which is different from the source domain (outdoor), moreover being only an unguided approach. Note that only transfer learning was performed on NYU v2 dataset with our model which had already been trained initially on SYNTHIA dataset in auxiliary learning setup with l_1 loss.

NYU dataset consists of RGB and depth images captured by Microsoft Kinect in indoor scenes. It provides 47,584 training images and 654 testing images, of dimensions 640×480. These images are reduced in dimensions by half and then center-cropped as performed by [25, 27]. The depth images were sparsified by uniformly sampling 500 points and used as input for the network. We followed the evaluation protocol of [25, 27], with error metrics as RMSE, REL (mean absolute relative error), and percentage of completed depth with a maximum of relative error and its inverse under a threshold t, where $t \in (1.25, 1.25^2, 1.25^3)$.

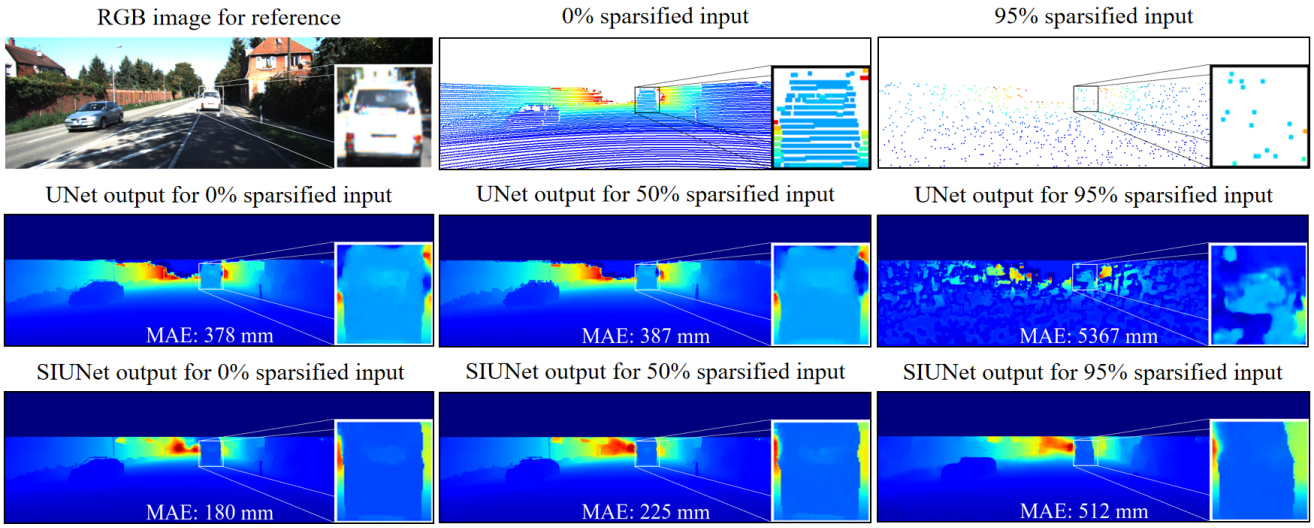


Figure 6: Sparsity invariance analysis: Sparse depth inputs were further randomly sparsified by values indicated in % on top of each reconstructed output. Indicated MAE (mm) values are an average over 1000 iterations. All the reconstructed images from our model have sharp object boundaries. Sparsified inputs have been dilated for visibility

Table 3: Numerical comparison of our approach with guided approaches on NYU v2 test set shows that our approach produces competitive results. Note that only transfer learning step (of Fig. 1) was performed with our model, which was initially trained on SYNTHIA dataset in auxiliary learning setup. ↓ indicates smaller is better

Method	Mode	RMSE (m) ↓	REL ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
Bilateral [38]	Guided	0.479	0.084	92.4	97.6	98.9
TGV [12]	Guided	0.635	0.123	81.9	93.0	96.8
Zhang *et al*. [51]	Guided	0.228	0.042	97.1	99.3	99.7
Ma *et al*. [27]	Guided	0.204	0.043	97.8	99.6	99.9
Nconv-CNN [10]	Guided	0.129	0.018	99.0	99.8	100
CSPN [4]	Guided	0.117	0.016	99.2	99.9	100
DeepLiDAR [33]	Guided	0.115	0.022	99.3	99.9	100
DepthNormal [45]	Guided	0.112	0.018	99.5	99.9	100
NLSPN [31]	Guided	0.092	0.012	99.6	99.9	100
SIUNet (Ours)	Unguided	0.138	0.015	99.2	99.8	100

5. Conclusion

In this paper, we propose sparsity invariant U-net architecture for unguided depth completion that relies only on LiDAR data of target domain during training and inference. It takes only sparse depth inputs, and produces dense depth images along with vestigial depth contours. Depth contour generation is an auxiliary task of our network (performed only on source domain) that enforces our network to learn structural information explicitly. Since both primary and auxiliary tasks are in depth domain, our network architecture is simple, and facilitates end-to-end sharing of features without branch-outs [50]. We showed that our network reconstructs dense depth images with consistent boundaries even for extremely sparse inputs. We demonstrated through qualitative and quantitative comparison on indoor and outdoor datasets, captured from different sensor systems, that our model achieves competitive performance among other unguided approaches and produces sharper object boundaries around depth discontinuity, despite having simple architecture.

Acknowledgement: This work has been funded by Fraunhofer-Gesellschaft, Germany.

References

[1] Jonathan T Barron and Ben Poole. The fast bilateral solver. In *ECCV*, pages 617–632, 2016.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE TPAMI*, 35(8):1798–1828, 2013.

[3] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, page 2147–2156, 2016.

[4] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *ECCV*, pages 103–119, 2018.

[5] Nathaniel Chodosh, Chaoyang Wang, and Simon Lucey. Deep convolutional compressed sensing for lidar depth completion. In *ACCV*, pages 499–513, 2018.

[6] Sarah Cwalina, Christoph Kottke, Volker Jungnickel, Ronald Freund, Patrick Runge, Pascal Rustige, Thomas Knieling, Shanshan Gu-Stoppel, Jörg Albers, Norman Laske, et al. Fiber-based frequency modulated lidar with mems scanning capability for long-range sensing in automotive applications. In *IEEE MetroAutomotive*, pages 48–53, 2021.

[7] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *CoRL*, pages 1–16, 2017.

[8] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Michael Persson. Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In *CVPR*, pages 12014–12023, 2020.

[9] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Propagating confidences through cnns for sparse data regression. *arXiv:1805.11913*, 2018.

[10] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE TPAMI*, 42:2423–2436, 2019.

[11] An Evgeniou and Massimiliano Pontil. Multi-task feature learning. *NeurIPS*, 19:41, 2007.

[12] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *ICCV*, pages 993–1000, 2013.

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32(11):1231–1237, 2013.

[14] Jiaqi Gu, Zhiyu Xiang, Yuwen Ye, and Lingxuan Wang. Denselidar: A real-time pseudo dense depth guided depth completion network. *IEEE RA-L*, 6:1808–1815, 2021.

[15] Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, and Marc Pollefeys. 3d visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *IJVC*, 68:14–27, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.

[17] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. *arXiv:2103.00783*, 2021.

[18] Zixuan Huang, Junming Fan, Shenggan Cheng, Shuai Yi, Xiaogang Wang, and Hongsheng Li. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *IEEE TIP*, 29:3429–3441, 2019.

[19] Chankyu Kim, Yunho Jung, and Seongjoo Lee. Fmcw lidar system to reduce hardware complexity and post-processing techniques to improve distance resolution. *Sensors*, 20:6676, 2020.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

[21] Bryan Krauss, Gregory Schroeder, Marko Gustke, and Ahmed Hussein. Deterministic guided lidar depth map completion. *arXiv:2106.07256*, 2021.

[22] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *IEEE CRV*, pages 16–22, 2018.

[23] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. *arXiv:1608.07916*, 2016.

[24] Lina Liu, Xibin Song, Xiaoyang Lyu, Junwei Diao, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Fcfr-net: Feature fusion based coarse-to-fine residual learning for depth completion. In *AAAI*, volume 35, pages 2136–2144, 2021.

[25] Kaiyue Lu, Nick Barnes, Saeed Anwar, and Liang Zheng. From depth what can you see? depth completion via auxiliary image reconstruction. In *CVPR*, pages 11306–11315, 2020.

[26] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *ICRA*, pages 3288–3295, 2019.

[27] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *ICRA*, pages 4796–4803, 2018.

[28] Jaritz Maximilian, De Charette Raoul, Wirbel Emilie, Perrotton Xavier, and Nashashibi Fawzi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *3DV*, pages 52–60, 2018.

[29] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE T-RO*, 33:1255–1262, 2017.

[30] Elizbar A Nadaraya. On estimating regression. *Theory Probab. Its Appl.*, 9:141–142, 1964.

[31] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *ECCV*, pages 120–136, 2020.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019.

[33] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for out-

door scene from sparse lidar data and single color image. In *CVPR*, pages 3313–3322, 2019.

[34] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, pages 12240–12249, 2019.

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[36] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, pages 3234–3243, 2016.

[37] Shreyas S Shivakumar, Ty Nguyen, Ian D Miller, Steven W Chen, Vijay Kumar, and Camillo J Taylor. Dfusenet: Deep fusion of rgb and sparse depth information for image guided dense depth completion. In *IEEE ITSC*, pages 13–20, 2019.

[38] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012.

[39] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[40] Hamid Hekma tian, Jingfu Jin, and Samir Al-Stouhi. Conf-net: Toward high-confidence dense 3d point-cloud with error-map prediction. *arXiv:1907.10148*, 2019.

[41] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *IEEE 3DV*, pages 11–20, 2017.

[42] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, , and Kilian Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, pages 8445–8453, 2019.

[43] Alex Wong, Safa Cicek, and Stefano Soatto. Learning topology from synthetic data for unsupervised depth completion. *IEEE RA-L*, 6:1495–1502, 2021.

[44] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *ICCV*, pages 2811–2820, 2019.

[45] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *ICCV*, pages 2811–2820, 2019.

[46] Lin Yan, Kai Liu, and Evgeny Belyaev. Revisiting sparsity invariant convolution: A network for image guided depth completion. *IEEE Access*, 8:126323–126332, 2020.

[47] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Baobei Xu, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. *arXiv:2107.13802*, 2021.

[48] Jing Yang, Bin Zhao, and Bo Liu. Distance and velocity measurement of coherent lidar based on chirp pulse compression. *Sensors*, 19:2313, 2019.

[49] Yanchao Yang, Alex Wong, and Stefano Soatto. Dense depth posterior (ddp) from single image and sparse range. In *CVPR*, pages 3353–3362, 2019.

[50] Jingwen Ye, Yixin Ji, Xinchao Wang, Kairi Ou, Dapeng Tao, and Mingli Song. Student becoming the master: Knowledge amalgamation for joint scene parsing, depth estimation, and more. In *CVPR*, pages 2829–2838, 2019.

[51] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *CVPR*, pages 175–185, 2018.

[52] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*, pages 4106–4115, 2019.

[53] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE TIP*, 2021.

[54] Yiming Zhao, Lin Bai, Ziming Zhang, and Xinming Huang. A surface geometry model for lidar depth completion. *IEEE RA-L*, 6:4457–4464, 2021.

[55] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *CVPR*, pages 8697–8710, 2018.