Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Sparsity Agnostic Depth Completion

(Article begins on next page)

06 May 2024

Sparsity Agnostic Depth Completion

Andrea Conti   Matteo Poggi   Stefano Mattoccia

Department of Computer Science and Engineering (DISI), University of Bologna, Italy

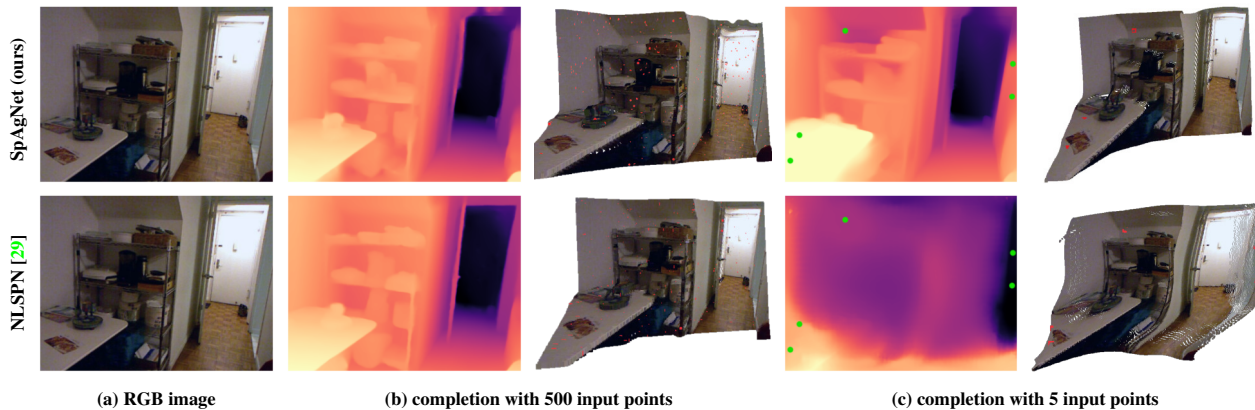Project page: https://andreaconti.github.io/projects/sparsity_agnostic_depth_completion

Figure 1: **Sparsity-agnostic depth completion.** From left to right: (a) reference image, (b) completed depth and point cloud using 500 depth points, (c) completed depth and point cloud using only 5 depth points. Our framework (top) dramatically outperforms NLSPN [29] (bottom) when both are trained with 500 points and tested with much fewer.

Abstract

We present a novel depth completion approach agnostic to the sparsity of depth points, that is very likely to vary in many practical applications. State-of-the-art approaches yield accurate results only when processing a specific density and distribution of input points, i.e. the one observed during training, narrowing their deployment in real use cases. On the contrary, our solution is robust to uneven distributions and extremely low densities never witnessed during training. Experimental results on standard indoor and outdoor benchmarks highlight the robustness of our framework, achieving accuracy comparable to state-of-the-art methods when tested with density and distribution equal to the training one while being much more accurate in the other cases. Our pretrained models and further material are available in our project page.

1. Introduction

Depth perception is pivotal to a variety of applications in robotics, scene understanding and more, and for this reason, it has been intensively investigated for decades. Among popular systems leveraging depth estimation, it is worth mentioning autonomous driving [9], path planning and aug-

mented reality. To date, accurate depth perception is demanded either to multi-view imaging approaches [44] or to specifically designed sensors such as ToF (Time of Flight) or LiDAR (Light Detection and Ranging). Although more expensive than standard cameras, depth sensors usually allow for higher accurate measurements even though at a lower spatial resolution. On the one hand, ToF sensors are cheap, small, and have been recently integrated into mobile consumer devices [16, 23]. They perturb the scene through coded signals unable to cope with outdoor daytime environments. To limit power consumption, a sparse emitting pattern is used, yielding meaningful depth measures for only a few points in the scene (∼500 points) [16]. On the other hand, LiDAR sensors employ a moving array of laser emitters scanning the scene and outputting a point cloud [33], which becomes a sparse depth map once projected over the image camera plane due to its much higher resolution. Devices leveraging such technology are expensive and bulky however, being applicable even in daylight outdoor environments, became standard for autonomous driving applications [37]. Since all these depth sensors provide – for different reasons – only sparse information, techniques aimed at recovering a dense depth map from an RGB image and a few measurements have gained much popularity in recent years [25, 29, 3].

Unfortunately, in real scenarios LiDAR and ToF sen-

sors are affected by additional issues other than sparsity, which may easily lead even to sparser depth points often unevenly distributed. For instance, the noise originating from multi-path interference – when multiple bouncing rays from different scene points collide on the same pixel – might lead the sensor to invalidate the measurement and consequently reduce density. Moreover, low-reflectivity surfaces/materials absorb the whole emitted signal while others reflect it massively, leading to saturation. Despite the two opposite behaviors, depth cannot be reliably measured in both cases, possibly leading to large, unobserved regions.

State-of-the-art depth completion techniques are fragile and fail at reconstructing the structure of the scene for areas where no depth points are available or when the sparsity changes significantly compared to the one used at training time. Indeed, the incapacity to deal with uneven spatial distributions of the sparse depth points – which will be unveiled in this work – threatens the possibility of deploying such solutions in different practical contexts. Moreover, this behaviour also prevents their seamless deployment when using a different sensor inferring the depth according to a spatial pattern different from the one used while training (e.g., switching from an expensive Velodyne [38] LiDAR system to a cheaper one).

Unfortunately, as reported in this paper and shown in Figure 1, convolutional layers struggle at generalizing when fed with variable sparsity input data. Hence, we propose a design strategy that diverges from the literature to overcome this issue by not directly feeding sparse depth points to the convolutional layers. Purposely, we iteratively merge the sparse input points with multiple depth maps predicted by the network. This strategy allows us to handle highly variable data sparsity, even training the network with a constant density distribution as done by state-of-the-art methods [29, 3, 7, 11] yet avoiding catastrophic drops in accuracy witnessed by competitors. Such an achievement makes our completion solution a *Sparsity Agnostic* Network, dubbed SpAgNet.

Our contribution can be summarized as follows:

- We propose a novel module designed to incorporate sparse data for depth completion yet being independent by their distribution and density. Such a module plugged into a competitive neural network architecture trained effortlessly can effectively deal with the previously mentioned issues.

- We assess the performance of SpAgNet and state-of-the-art methods on a set of highly challenging cases using KITTI Depth Completion (DC) and NYU Depth V2 (NYU) datasets. We highlight the superior robustness of our solution compared to state-of-the-art when dealing with uneven input patterns.

2. Related Work

Depth Prediction. Except for a few attempts to solve monocular depth prediction through non-parametric approaches [17], the practical ability to solve this ill-posed problem has been achieved only with the deep learning revolution. At first, deploying plain convolutional neural networks [6] and then, through more complex approaches. Specifically, [8] casts the problem as a classification task, [1] exploits a bidirectional attention mechanism, [19] introduces novel local planar guidance layers to better perform the decoding phase, [32] jointly computes panoptic segmentation to improve depth prediction performance, [34] unifies multiple depth sources to coherently train a neural network to better generalize. The previous methods require a massive quantity of training data to achieve proper performance in unknown environments thus self-supervised paradigms gained much attention. For instance, [10] relies on a supervisory signal extracted from a monocular video stream.

Depth Completion. Depth completion aims at densifying the sparse depth map obtained by an active depth sensor, providing sparser to denser measurements depending on the technology – e.g., as evident by comparing Radar [20] versus LiDAR [9] sensors. This task has been tackled either by leveraging an additional RGB image or barely using the sparse depth data. Although most methods rely on learning-based paradigms, [45] proposes a non-parametric handcrafted method. Regarding deep-learning methods, [25] was among the first to tackle the problem by jointly feeding a neural network with the RGB frame and the sparse depth points to densify the latter. Observing that manipulating sparse data is sub-optimal for convolutions, [37, 4] proposed custom convolutional layers explicitly taking into account sparsity. Eventually, guided spatial propagation techniques have demonstrated superior performance. At first, [21] proposed a network able to learn local affinities to guide the depth expansion, this strategy was improved initially by [3] and then by [29]. Based on a similar principle, [36] proposes content-dependent and spatially-variant kernels for multi-modal feature fusion. [7] performs depth completion also modeling the confidence of the sparse input depth and the densified output. In a parallel track, a few works focused on unsupervised training strategies for depth completion [24, 41, 40, 42]. Finally, [11] proposes an approach to deal with depth completion and depth prediction. Even though this seems similar to our research, it is only loosely related. First, it cannot deal with different sparsities but only with the total absence of the sparse depth points. Second, to achieve their goal, they need a specific training procedure and an additional branch to handle the availability of sparse depth data. In contrast, our peculiar network design addresses both issues.

Uncertainty Estimation. Evaluating the estimated value's uncertainty (or confidence) is essential in many cir-
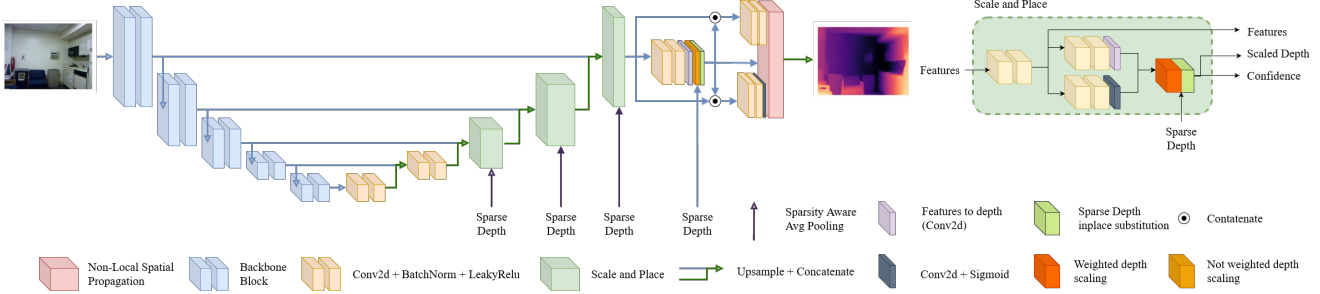
Figure 2: **SpAgNet architecture.** The network follows an encoder-decoder design, with a backbone to extract features from the image and a custom decoder to iteratively merge at multiple scales sparse depth hints without directly feeding them as a sparse depth map. Finally, we leverage non-local propagation [29] to improve accuracy further.

cumstances. For neural networks, it has been widely explored either the use of Bayesian frameworks [26, 39, 2] or strategies jointly predicting the mean and variance of the network's output distribution [28]. For depth completion, [7] proposed to jointly compute the confidence of the sparse input depth and of the densified output. While, for monocular depth prediction, [31] has deeply investigated uncertainty for self-supervised approaches.

3. Sparsity Agnostic Framework

To tackle depth completion, we start from our previous observations. Specifically, as pointed out by [37, 4], 2D convolutions struggle to manipulate sparse information. Additionally, we further notice that the density of such input depth data and its spatial distribution – which could be highly uneven – might lead state-of-the-art networks to catastrophic failures, as depicted at the bottom of Figure 1. Moreover, we argue that these networks mostly rely on the sparse depth input overlooking the image content substantially ignoring the geometric structure depicted in it.

SpAgNet relies on an encoder-decoder structure with skip connections, as depicted in Figure 2. However, unlike current depth completion techniques [29, 3, 11, 7], we do not feed the encoder with sparse depth information for the reasons previously outlined. We extract instead features from the RGB frame *only* in order to get rid of the sparse input data and, consequently, its density. This strategy allows us to constrain the network to exploit the image content fully and, as we will discuss later, to enforces the network to extract the geometry of the scene from RGB.

The decoding step predicts – iteratively and at multiple scales – dense depth from the RGB image and *fuse* it with the sparse input data. The first iterative step takes the input features extracted from the RGB image and generates a lower scale depth map and a confidence map. Then, the next iterative steps process the same inputs plus the depth map and its confidence, both *augmented* with the sparse input points computed in the previous iteration. Moreover, since

each intermediate depth map provides information up to a scale factor, we scale it according to the sparse input points before each augmenting step. We do so due to the ill-posed nature of monocular depth prediction. Experimental results will corroborate our design choice, especially when dealing with a few sparse input points. At the end of the iterative steps, we apply the non-local spatial propagation module proposed in [29] to refine the depth map inferred by the network. Figure 2 describes the whole framework.

3.1. Encoder Architecture

Since our framework encodes features from the image only, we can leverage as encoding backbone any pretrained network. Such backbone is pre-trained on ImageNet [35]. Among the multiple choices [12, 14, 43] we choose ResNeXt50 [43] due to its good trade-off between performance and speed. Specifically, it downsamples the image to scales $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ and $\frac{1}{32}$ and the features used in the decoding step as input and skip connection.

3.2. Scale and Place Module

In our proposal, the core Scale and Place (S&P) module is in charge of inferring a dense and scaled depth map and its confidence. It takes as input the backbone features, the output of the previous S&P module at a different scale, and the sparse depth points as depicted in Figure 2.

Specifically, S&P leverages the input features to jointly generate an initial up-to-scale depth map and its confidence deploying a stem block composed of two convolutional layers and two heads in charge of generating them. Each convolutional layer consists of a 2D convolution, a batch normalization [15] and a Leaky ReLU. Then in the *Scale* step, the S&P module performs a weighted linear regression to scale the depth map according to the available sparse input points, weighted by means of confidence. The parameters of the weighted linear regression can be computed in closed form and in a differentiable way, as described in Eq. 1 where p_i is the predicted depth value and c_i its confidence

(a) RGB   (b) dense depth

(c) confidence map   (d) depth error
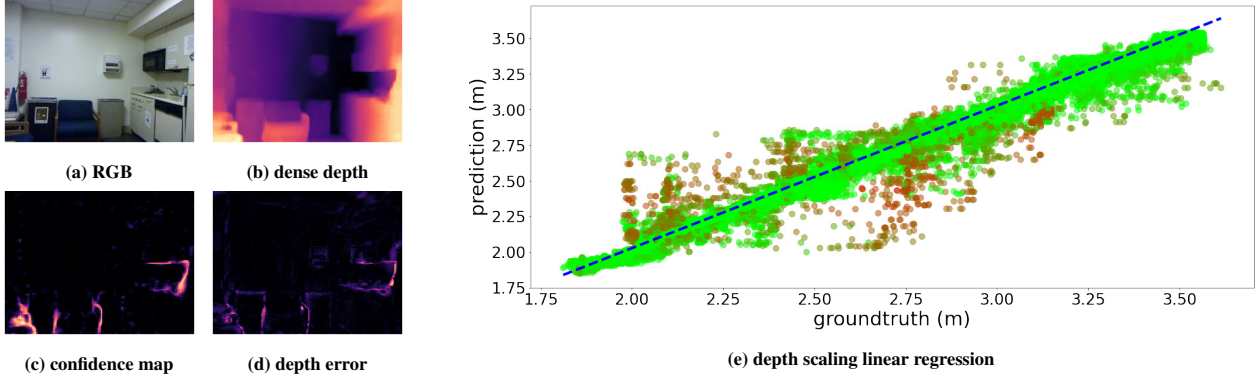
(e) depth scaling linear regression

Figure 3: **Confidence aware depth scaling.** Example of confidence usage to scale depth. On left, we show (a) the input image, (b) predicted depth map, (c) estimated confidence and (d) errors with respect to groundtruth. On right, we plot the outcome of the scaling procedure (red means a lower confidence prediction, green a higher one).

corresponding to an available input sparse point s_i.

$$\beta = \frac{\sum_i c_i(p_i - \hat{p})(s_i - \hat{s})}{\sum_i c_i(p_i - \hat{p})^2} \quad \alpha = \hat{s} - \beta\hat{p} \quad (1)$$

$$\hat{p} = \frac{\sum_i c_i p_i}{\sum_i c_i} \quad \hat{s} = \frac{\sum_i c_i s_i}{\sum_i c_i}$$

Then, in the *Place* step, for those points where a sparse input depth value is available, we replace the corresponding value in the scaled depth map with it. Additionally, we update the same point in the confidence map with the highest score. The latter step can be summarized as follows

$$\hat{D}[x,y] = \begin{cases} D^s[x,y] & \text{if} \quad H[x,y] = 0 \\ H[x,y] & \text{if} \quad H[x,y] \neq 0 \end{cases} \quad (2)$$

$$\hat{C}[x,y] = \begin{cases} C^s[x,y] & \text{if} \quad H[x,y] = 0 \\ 1 & \text{if} \quad H[x,y] \neq 0 \end{cases} \quad (3)$$

where D^s is the scaled depth map, C^s is the confidence map and H is a sparse depth map containing zeros where an input sparse depth point is not available. The predicted confidence has an empirically chosen range of [0.1 .. 0.9] while we associate confidence 1 to each valid value in H.

We apply the S&P module at scales $\frac{1}{8}$, $\frac{1}{4}$ and $\frac{1}{2}$. The module at $\frac{1}{8}$ computes the initial depth and confidence maps leveraging only the RGB features. The others take in input also the up-sampled dense depth and confidence maps from the previous module in order to iteratively correct the prediction relying on both the predicted depth and the injected sparse points. Thus, with this strategy, the decoder does not deal directly with sparse data in any of its steps. Nonetheless, the network can locate and effectively leverage reliable sparse information. An example of this mechanism is

showed in Figure 3, where can be clearly seen how the network learns to locate the most reliable depth values as those closer to the groundtruth depth.

It is worth noting that confidence plays a crucial role in the S&P module. At first, in the *Scale* step, it helps to locate outliers in the estimated depth map enabling to soften their impact when performing the scaling procedure. Additionally, in the *Place* step, assigning the highest confidence to the sparse input points enables the network to rely on them effectively. Nonetheless, SpAgNet also exploits the other predicted depth points according to their estimated confidence.

Since the S&P module needs the sparse data at multiple scales, we down-sample it by employing a non-parametric sparsity aware pooling: moving a 3×3 window with stride 2, we assign the mean of the available measures in its neighbourhood to each coordinate, we iteratively apply this process to reach lower resolutions. This approach leads to a densification of the sparse depth map and helps, at all scales, to include even the meagre few sparse points available to a large field of view.

3.3. Non-Local Spatial Propagation

Spatial propagation concerns the diffusion of information in a localized position to its neighbourhoods. This strategy represents a common practice in the depth completion literature [21, 3, 29, 13] and can be achieved by a neural network in charge of learning the affinity among neighbours. Let $X = (x_{m,n}) \in R^{M \times N}$ be a 2D depth map to be refined through propagation, at step t it acts as follows:

$$x_{m,n}^t = w_{m,n}^c x_{m,n}^{t-1} + \sum_{(i,j) \in N_{m,n}} w_{m,n}^{i,j} x_{i,j}^{t-1} \quad (4)$$

Where (m, n) is the reference pixel currently being updated, $(i,j) \in N_{m,n}$ the coordinate of the pixels in its

neighborhood, $w_{m,n}^{i,j}$ the affinity weights, and $w_{m,n}^{c}$ the affinity weight of the reference pixel:

$$w_{m,n}^{c} = 1 - \sum_{(i,j) \in N_{m,n}} w_{m,n}^{i,j} \qquad (5)$$

The various existing methods differ by the choice of the neighborhood and by the normalization procedure of the affinity weights, the latter necessary to ensure stability during propagation [21, 3, 29]. Within SpAgNet, we implement the non-local approach [29], letting the network dynamically decide the neighborhood using deformable convolutions [5]. Formally:

$$N_{m,n} = \{x_{m+p,n+q} \mid (p,q) \in f_\phi(I, H, n, m)\} \qquad (6)$$
$$p, q \in \mathbb{R}$$

Where I and H are the RGB image and the sparse depth, and $f_\phi(\cdot)$ is the neural network determining the neighbourhood. The non-local propagation module requires in input an initial depth map generated through two convolutional blocks from the last S&P block output, scaled using the full-resolution sparse depth points. However, in this case, we do not perform a weighted scaling to obtain the best result on the entire frame. Finally, as usual, the sparse depth points override the predicted output. The resulting depth map is then fed along with features to two convolutional blocks to generate the guiding features and confidence required by the propagation module.

3.4. Loss Function

At each scale, we train the network by supervising the depth obtained by the S&P module before *Place* step. The confidence weights the loss of each depth prediction, and a regularization term (controlled by η) enforces the network to maintain the confidence as higher as possible. Following [29], we compute both L1 and L2 losses. Our loss function, at a specific scale, is described by Eq. 7 where C^s and D^s are respectively confidence and depth at a specific scale s. Confidence is not computed for the full size scale, hence $C^0 = 1$. Finally, it is worth mentioning that lower scales are weighted less through an exponential decay factor γ.

$$L = \sum_{s=0}^{n} \gamma^s \frac{1}{N} \sum_{i}^{m} C_i^s L_i^{12} - \eta \ln C_i^s \qquad (7)$$
$$L_i^{12} = |D_i^s - G_i| + |D_i^s - G_i|^2$$

4. Experimental Results

We have implemented SpAgNet in PyTorch [30] training with 2 NVIDIA RTX 3090 and using the ADAM optimizer [18] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The final model requires 35 milliseconds to perform a prediction on a image

of 640×480 resolution employing a single NVIDIA RTX 3090 GPU.

4.1. Datasets

NYU Depth V2. The NYU Depth V2 [27] dataset is an indoor dataset containing 464 indoor scenes gathered with a Kinect sensor. We follow the official train/test split as previous works relying on the pre-processed subset by Ma et al. [25] using 249 scenes for training (∼50K samples) and 215 scenes (654 samples) for testing. Each image has been down-sampled to 320×240 and then center cropped to 304×228. As a common practice on this dataset, 500 random points per image have been extracted to simulate sparse depth. We train our network for 15 epochs starting with a learning rate 10^{-3} and decreasing it every 3 epochs by 0.1, setting $\gamma = 0.4$ and $\eta = 0.1$. We use batch size 24 (12 for each GPU); hence the network is extremely fast to converge since the whole training accounts less than 30K steps. We apply color and brightness jittering and horizontal flips to limit overfitting.

KITTI Depth Completion (DC). KITTI DC [37] is an outdoor dataset containing over 90K samples, each one providing RGB information and aligned sparse depth information (with a density of about 5%) retrieved by a high-end Velodyne HDL-64E LiDAR sensor. The images have 1216×352 resolution, and the dataset provides a standard split to train (86K samples), validate (7K samples) and test (1K samples). The groundtruth has been obtained temporally accumulating multiple LiDAR frames and filtering errors [37], leading to a final density of about 20%. On this dataset we train for 10 epochs with batch size 8 (4 for each GPU), starting with learning rate 10^{-3} and we decrease it every 3 epochs by 0.1, we set $\gamma = 0.4$ and $\eta = 20.0$. Data augmentation follows the same scheme used for NYU.

4.2. Evaluation

In this section, we assess the performance of our proposal and state-of-the-art methods deploying the dataset mentioned above. Following standard practice [29, 3], we use the following metrics: RMSE $= \sqrt{\frac{1}{N} \sum_i |D_i - G_i|^2}$, MAE $= \frac{1}{N} \sum_i |D_i - G_i|$ and REL $= \frac{1}{N} \sum_i \left| \frac{D_i - G_i}{G_i} \right|$.

For evaluation purposes, in addition to the standard protocol deployed in this field [29, 3], we also thoroughly evaluate the robustness of the networks on the two datasets in much more challenging scenarios but always training with the standard procedure (i.e., using 500 points on NYU and 64 LiDAR lines on KITTI). Since KITTI DC is thought for autonomous driving tasks and the sparse depth is acquired with an high end 64 Lines Lidar which provides in output always the same pattern, we simulate the switch to a cheaper device providing in output less lines assessing the capability of SpAgNet to generalize over sparse depth density. On
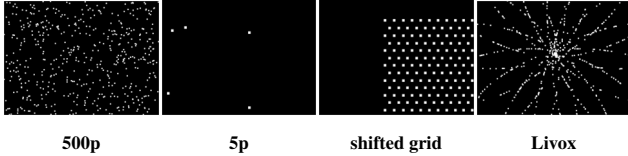
| 500p | 5p | shifted grid | Livox |

Figure 4: **Sparse depth patterns.** Examples of different sparse depth patterns, from left to right: 500 random points, 5 random points, shifted triangular tiling dot pattern and a Livox-like pattern (e.g. Livox Mid-70).

NYU Depth V2, sparse depth points are traditionally extracted randomly from the groundtruth [25, 29, 3] which is almost dense. Thus, we test i) the extreme case of having only 5 random points, ii) the impact of having large empty areas and iii) the impact of changing the sparsity pattern. We implement ii) sampling from the groundtruth a triangular tiling dot pattern aimed at simulating the output of a commercial VCSEL [23] ToF sensor and then randomly shifting this pattern to leave behind large empty areas where no sparse hints are available while iii) extracting from the groundtruth sparse points with the pattern of a Livox Mid-70 [22]. All these patterns are showed in Figure 4. We take into account the publicly pre-trained state-of-the-art models available either on NYU Depth V2 or KITTI DC and we take care to guarantee that each architecture sees exactly the *same* sparse points while being evaluated.

Results on NYU Depth v2. Table 1 compares state-of-art methods and our proposal on the NYU dataset using different input configurations: in the upper portion by changing the number of samples and in the lower portion by changing the pattern type. From the table, we can notice that our proposal achieves competitive results, being very close to NLSPN and better than other methods when the number of points used is the same as the training phase (i.e., 500). Similar behaviour occurs with 200 points. However, when the density of input points decreases further, SpAgNet vastly outperforms the state-of-the-art. The performance gap with other methods gets much higher when decreasing the density further. For instance, with 50 points, the RMSE by SpAgNet is 0.272 m, while the second one (NLSPN) accounts for 0.423 m. Notably, with only 5 points, the same metrics are 0.467 m and 1.033 m (NLSPN), further emphasizing the ability of our proposal to deal even with meagre input points, in contrast to our competitors. It is worth observing that our method outperforms competitors with randomly selected input points starting from 100.

The bottom portion of Table 1 reports the outcome of the evaluation with different spatial distributions and their average density of depth input points. Specifically, we report results with the two distributions depicted in the rightmost images of Figure 4. From the table, we can observe that when the spatial distribution covers the whole image,

Method	Samples	REL↓	RMSE (m)↓
pNCNN [7]		0.026	0.170
CSPN [3]		0.016	0.118
NLSPN [29]	500	**0.013**	**0.101**
PackNet-SAN [11]		0.019	0.120
SpAgNet (ours)		0.015	0.114
pNCNN [7]		0.040	0.237
CSPN [3]		0.027	0.177
NLSPN [29]	200	**0.019**	**0.142**
PackNet-SAN [11]		0.027	0.155
SpAgNet (ours)		0.024	0.155
pNCNN [7]		0.061	0.338
CSPN [3]	100	0.067	0.388
NLSPN [29]		**0.038**	0.246
SpAgNet (ours)		**0.038**	**0.209**
pNCNN [7]		0.108	0.568
CSPN [3]	50	0.185	0.884
NLSPN [29]		0.081	0.423
SpAgNet (ours)		**0.058**	**0.272**
pNCNN [7]		0.722	2.412
CSPN [3]	5	0.581	2.063
NLSPN [29]		0.262	1.033
SpAgNet (ours)		**0.131**	**0.467**
pNCNN [7]		0.519	1.922
CSPN [3]	shifted grid	0.367	1.547
NLSPN[29]	(∼ 100)	0.175	0.796
SpAgNet (ours)		**0.110**	**0.422**
pNCNN [7]		0.061	0.333
CSPN [3]	livox	0.066	0.376
NLSPN [29]	(∼ 150)	**0.037**	0.233
SpAgNet (ours)		0.039	**0.206**

Table 1: **Evaluation on NYU Depth v2.** Comparison with state-of-the-art methods, trained with 500 random points, extracted from groundtruth, as input and tested with different densities and patterns. In bold is the best result, underlined the second one.

as in the case of the Livox-like pattern, SpAgNet and NLSPN achieve similar performance while other methods fall behind. However, when the input points do not cover significant portions of the scene and the density decreases further, like in the shifted-grid case, our method dramatically outperforms all competitors by a large margin.

Figure 5 shows qualitatively how SpAgNet compares to CSPN and NLSPN on an NYU sample when using 500 random points, 5 points and the shifted grid. It highlights how only our method yields meaningful and compelling results with 5 points and the shifted grid, leveraging the image content much better than competitors, thanks to the proposed architectural design. At the same time, our network achieves results comparable to competitors with 500 randomly distributed points. This fact further highlights that the robustness of SpAgNet is traded with the capacity of entirely leveraging the sparse depth information when fully available.

Results on KITTI DC. Once we assessed the performance on the indoor NYU dataset, we report in Table 2 the evaluation on KITTI DC. From the table, we can notice that with 64 lines, SpAgNet results almost comparable to the best one, NLSPN. However, by reducing the number of
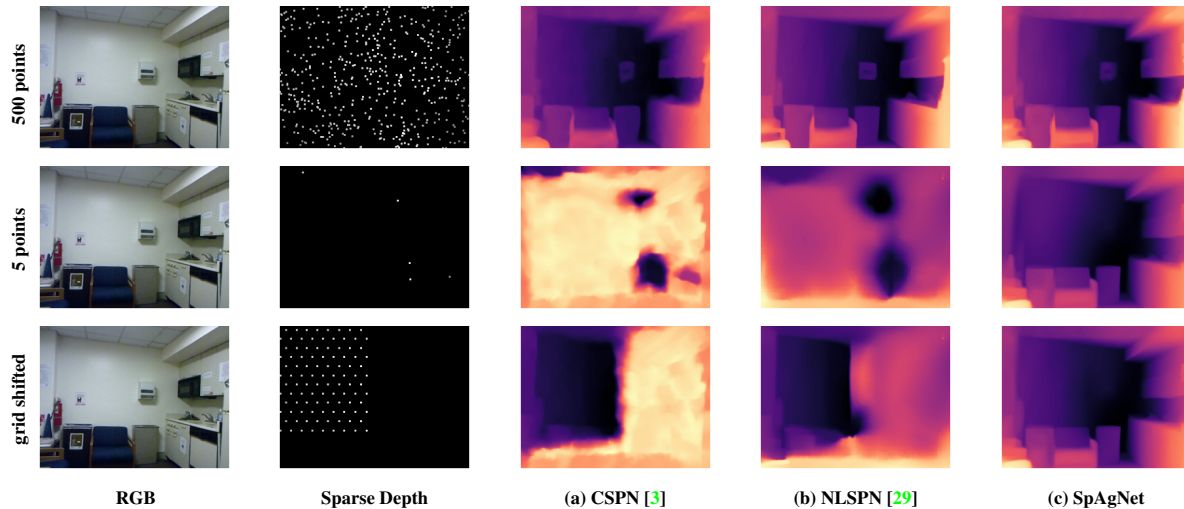
Figure 5: **Qualitative results on NYU-Depth v2.** CSPN and NLSPN, when processing 5 points or the shifted grid pattern, manifest the complete inability to handle them, while SpAgNet maintains the scene structure.

Method	Lines	RMSE (mm) ↓	MAE ↓
NLSPN [29]		**778.00**	**199.50**
pNCNN [7]		1011.86	255.93
PackNet-SAN [11]	64	1027.32	356.04
PENet [13]		<u>791.62</u>	242.25
SpAgNet (ours)		844.79	<u>218.39</u>
NLSPN [29]		<u>1217.21</u>	<u>367.49</u>
pNCNN [7]		1766.84	615.93
PackNet-SAN [11]	32	1836.84	914.33
PENet [13]		1853.06	1025.42
SpAgNet (ours)		**1164.18**	**339.22**
NLSPN [29]		<u>1988.52</u>	<u>693.10</u>
pNCNN [7]		3194.69	1321.74
PackNet-SAN [11]	16	2841.35	1570.05
PENet [13]		3538.02	2121.46
SpAgNet (ours)		**1863.25**	**606.92**
NLSPN [29]		3234.93	<u>1491.28</u>
pNCNN [7]		5921.94	2999.92
PackNet-SAN [11]	8	<u>3231.03</u>	1575.41
PENet [13]		6015.02	3812.45
SpAgNet (ours)		**2691.34**	**1087.21**
NLSPN [29]		<u>4834.22</u>	2742.80
pNCNN [7]		9364.58	5362.45
PackNet-SAN [11]	4	4850.20	<u>2255.08</u>
PENet [13]		9318.86	5819.36
SpAgNet (ours)		**3533.74**	**1622.64**

Table 2: **Evaluation on KITTI DC.** Comparison with state-of-the-art methods, always trained on 64 lines Velodyne lidar and tested with a different number of lines. In bold is the best result, underlined the second one.

lines from 32 to 4, our network gets always the best performance with an increasing gap. Interestingly, PackNet-SAN [11], which has been specifically trained to perform well in both depth completion (64 lines) and depth prediction (0 lines) is not able to deal with fewer lines. Indeed, the accuracy it achieves when processing 16, 8 or 4 lines is even lower than the one achieved when performing depth prediction, i.e. with RMSE equal to 2.233 mm. We ascribe this

behaviour to the fact that they train an external encoding branch to extract features from sparse data and feed them to the network by means of a sum operation. Even though such a branch applies a special and bulky sparse convolution operator [4], it does not seem capable of generalizing to fewer points. On the contrary, the whole network seems to suffer of the same issues of fully convolutional models, resulting effective only when fed with 64 LiDAR lines or none – the only configurations observed during training.

Figure 6 shows, on an image of the KITTI DC dataset and for three different numbers of lines, the outcome of NLSPN, PENet and our network. In contrast to competitors, SpAgNet consistently infers meaningful depth maps, even when the number of lines decreases. This behaviour can be perceived better by looking at the error maps. For instance, it is particularly evident with 4 lines, focusing on the road surface and the far and background objects.

Additional qualitative results are reported as videos in the **supplementary material** and in our project page.

4.3. Ablation Study

Finally, we carry out an ablation study concerning the main components of SpAgNet to measure their effectiveness. Specifically, in Table 3, we conduct two main studies, respectively, to evaluate (a) the impact of i) the Scale step of the S&P modules (while the Place step is strictly necessary, being it the entry point to input the sparse depth points needed to perform completion), ii) the usage of confidence and iii) the non-local propagation head, and (b) results achieve with different backbones.

From (a), we can notice that with 500 sparse points, scaling does not significantly improve since the network already learns to generate an output that is almost in scale. How-
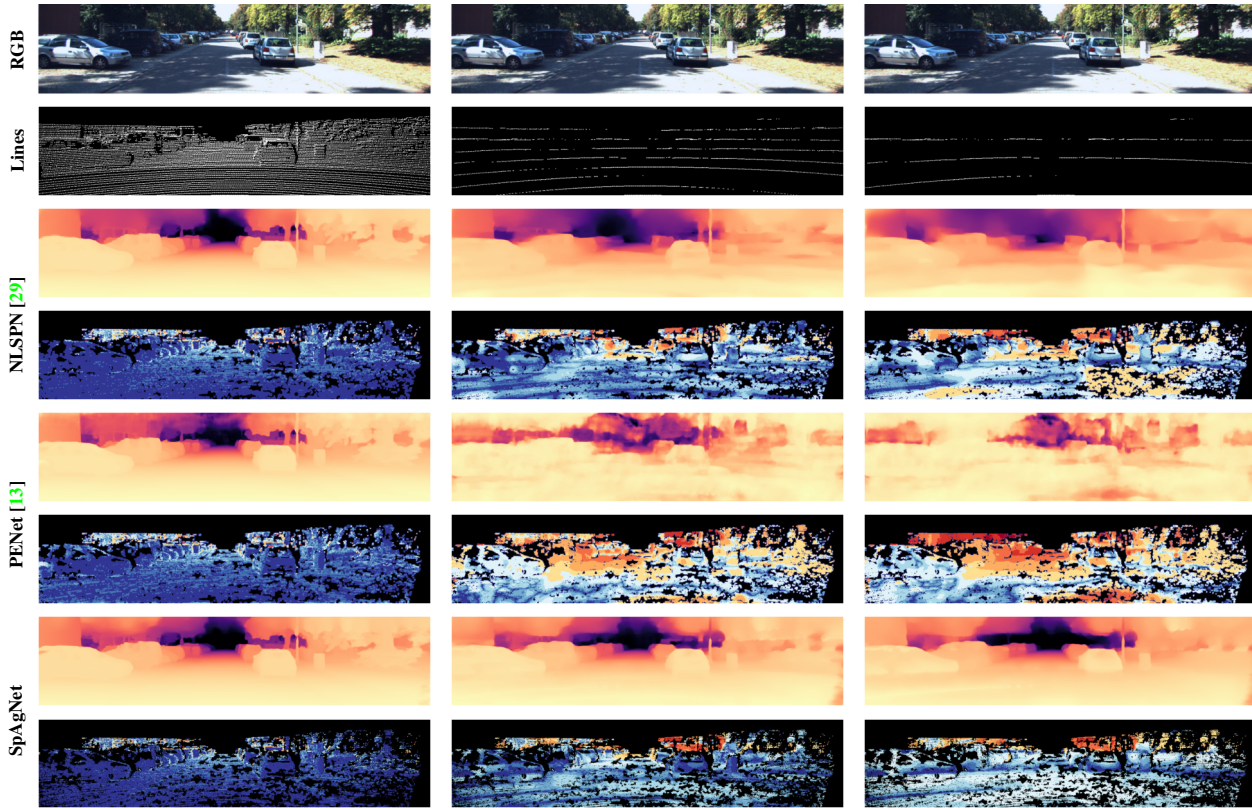
Figure 6: **Qualitative results on KITTI DC.** We report results using, respectively, from left to right, 64, 8 and 4 lines. From top to bottom the predicted depth and error map of [29], [13] and ours.

(a)

NLSP	Confidence	Scaling	Samples	RMSE (m) ↓
✗	✗	✗		0.161
✗	✓	✓		0.127
✓	✗	✓		0.122
✓	✓	✗		0.115
✓	✗	✗	500	0.132
✗	✓	✗		0.145
✗	✗	✓		0.135
✓	✓	✓		**0.114**
✗	✗	✗		0.770
✗	✗	✓		0.474
✓	✗	✓		0.479
✓	✓	✗		0.526
✓	✗	✗	5	0.566
✗	✓	✗		0.823
✗	✗	✓		0.484
✓	✓	✓		**0.467**

(b)

Backbone	Size	Samples	RMSE (m) ↓
ResNet18	27M		0.116
ResNet34	37M		0.121
ResNet50	51M	500	0.117
ResNeXt50	51M		**0.114**
DenseNet121	30M		0.118
DenseNet161	61M		0.115
ResNet18	27M		0.504
ResNet34	37M		0.474
ResNet50	51M	5	0.664
ResNeXt50	51M		**0.467**
DenseNet121	30M		0.678
DenseNet161	61M		0.564

Table 3: **Ablation study on NYU – (a) single components, (b) different backbones.** Training with 500 points, testing either with 500 or 5 points on the same dataset.

tested with 500 points, with ResNeXt50 achieving slightly better results. A significant gap in accuracy emerges when testing the same networks with only 5 points, with ResNeXt50 achieving the best results again.

5. Conclusion

This paper proposes a sparsity agnostic framework for depth completion relying on a novel Scale and Place (S&P) module. Injecting sparse depth points to it rather than to convolutions allows us to improve the robustness of the architecture even when facing uneven and sparse distributions of input depth points. In contrast, existing state-of-the-art solutions are not robust in such circumstances and are often unable to infer meaningful results. Experimental results demonstrate the ability of our network to be competitive with state-of-the-art facing standard input distributions, while resulting much better when dealing with uneven ones.

ever, with only 5 points, applying a global scaling procedure helps retrieve the correct scale even in regions lacking depth measurements. Focusing on confidence, it turns out to be effective with high and low densities of input points. Finally, Non-Local Spatial Propagation further boosts performance in both cases.

In (b), most backbones yield comparable results when

References

[1] Shubhra Aich, Jean Marie Uwabeza Vianney, Md Amirul Islam, and Mannat Kaur Bingbing Liu. Bidirectional attention network for monocular depth estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11746–11752, 2021.

[2] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1683–1691, Bejing, China, 22–24 Jun 2014. PMLR.

[3] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018.

[4] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[6] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[7] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Michael Persson. Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12011–12020, 2020.

[8] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[10] Clement Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3827–3837, United States, Feb. 2020. Institute of Electrical and Electronics Engineers (IEEE). International Conference on Computer Vision 2019, ICCV 2019 ; Conference date: 27-10-2019 Through 02-11-2019.

[11] Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[13] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13656–13662, 2021.

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[16] Xiaowen Jiang, Valerio Cambareri, Gianluca Agresti, Cynthia I Ugwu, Adriano Simonetto, Pietro Zanuttigh, and Fabien Cardinaux. A Low Memory Footprint Quantized Neural Network for Depth Completion of Very Sparse Time-of-Flight Depth Maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2022.

[17] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2144–2158, 2014.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[19] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation, 2019.

[20] Juan-Ting Lin, Dengxin Dai, and Luc Van Gool. Depth estimation from monocular images and sparse radar data. In *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[21] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[22] Livox Technology Company Limited. https://www.livoxtech.com/mid-70.

[23] Gregor Luetzenburg, Aart Kroon, and Anders Bjørk. Evaluation of the apple iphone 12 pro lidar for an application in geosciences. *Scientific Reports*, 11, 11 2021.

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

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

[26] David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

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

[28] D.A. Nix and A.S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 1, pages 55–60 vol.1, 1994.

[29] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *European Conference on Computer Vision*, pages 120–136. Springer, Jul 2020.

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, 2017.

[31] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3224–3234, 2020.

[32] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3997–4008, June 2021.

[33] Thinal Raj, Fazida Hanim Hashim, Aqilah Baseri Huddin, Mohd Faisal Ibrahim, and Aini Hussain. A survey on lidar scanning mechanisms. *Electronics*, 9(5), 2020.

[34] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Agost 2020.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[36] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30:1116–1129, 2020.

[37] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20, 2017.

[38] Velodyne Lidar. https://velodynelidar.com/products/puck.

[39] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress.

[40] Alex Wong, Safa Cicek, and Stefano Soatto. Learning topology from synthetic data for unsupervised depth completion. *IEEE Robotics and Automation Letters*, 6(2):1495–1502, 2021.

[41] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5(2):1899–1906, 2020.

[42] Alex Wong and Stefano Soatto. Unsupervised depth completion with calibrated backprojection layers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12747–12756, 2021.

[43] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017.

[44] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[45] Yiming Zhao, Lin Bai, Ziming Zhang, and Xinming Huang. A surface geometry model for lidar depth completion. *IEEE Robotics and Automation Letters*, 6(3):4457–4464, 2021.