

# DeCoTR: Enhancing Depth Completion with 2D and 3D Attentions

Yunxiao Shi Manish Kumar Singh Hong Cai Fatih Porikli  
Qualcomm AI Research\*

{yunxshi, masi, hongcai, fporikli}@qti.qualcomm.com

## Abstract

*In this paper, we introduce a novel approach that harnesses both 2D and 3D attentions to enable highly accurate depth completion without requiring iterative spatial propagations. Specifically, we first enhance a baseline convolutional depth completion model by applying attention to 2D features in the bottleneck and skip connections. This effectively improves the performance of this simple network and sets it on par with the latest, complex transformer-based models. Leveraging the initial depths and features from this network, we uplift the 2D features to form a 3D point cloud and construct a 3D point transformer to process it, allowing the model to explicitly learn and exploit 3D geometric features. In addition, we propose normalization techniques to process the point cloud, which improves learning and leads to better accuracy than directly using point transformers off the shelf. Furthermore, we incorporate global attention on downsampled point cloud features, which enables long-range context while still being computationally feasible. We evaluate our method, DeCoTR, on established depth completion benchmarks, including NYU Depth V2 and KITTI, showcasing that it sets new state-of-the-art performance. We further conduct zero-shot evaluations on ScanNet and DDAD benchmarks and demonstrate that DeCoTR has superior generalizability compared to existing approaches.*

## 1. Introduction

Depth is crucial for 3D perception in various downstream applications, such as autonomous driving, augmented and virtual reality, and robotics [1, 2, 8, 9, 11, 33–35, 43, 50, 51]. However, sensor-based depth measurement is far from perfect. Such measurements often exhibit sparsity, low resolution, noise interference, and incompleteness. Various factors, including environmental conditions, motion, sensor power constraints, and the presence of specular, transparent, wet, or non-reflective surfaces, contribute to these limitations. Consequently, the task of depth completion, aimed at

generating dense and accurate depth maps from sparse measurements alongside aligned camera images, has emerged as a pivotal research area [4, 5, 24, 26, 27, 29, 41, 45].

Thanks to the advances in deep learning, there has been significant progress in depth completion. Earlier papers leverage convolutional neural networks to perform depth completion with image guidance and achieve promising results [3, 27, 37]. In order to improve accuracy, researchers have studied various spatial propagation methods [4, 24, 25, 29], which performs further iterative processing on top of depth maps and features computed by an initial network. Most existing solutions build on this in the last stage of their depth completion pipeline to improve performance [17, 45]. These propagation algorithms, however, focus on 2D feature processing and do not fully exploit the 3D nature of the problem. A few recent papers utilize transformers for depth completion [32, 45]. However, they apply transformer operations mainly to improve feature learning on the 2D image plane and fail to achieve acceptable accuracy without employing spatial propagation.

Several studies have looked into harnessing 3D representation more comprehensively. For instance, [18, 49] construct a point cloud from the input sparse depth, yet coping with extreme sparsity poses challenges in effective feature learning. Another approach, as seen in [26], uplifts 2D features to 3D by using the initial dense depth predicted by a simple convolutional network, but it is impeded by the poor accuracy of the initial network and requires dynamic propagations to attain acceptable accuracy. Very recently, researchers have proposed employing transformers for 3D feature learning in depth completion [44]; however, this work applies transformer layers to extremely sparse points, which is ineffective for learning informative 3D features.

Here, we introduce DeCoTR to perform feature learning in full 3D. It accomplishes this by constructing a dense feature point cloud derived from completed depth values obtained from an initial network and subsequently applying transformer processing to these 3D points. To do this properly, it is essential to have reasonably accurate initial depths. As such, we first enhance a commonly used convolution-based initial depth network, S2D [27], by integrating trans-

\* Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

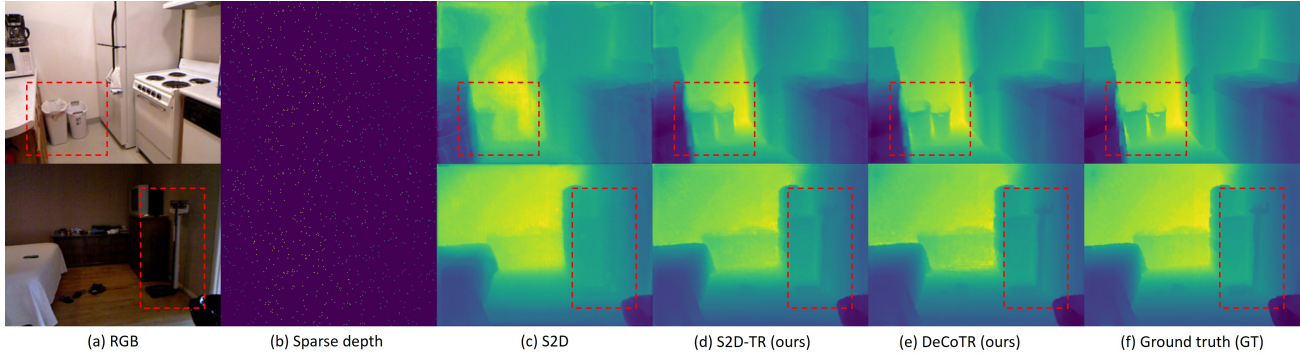


Figure 1. Example depth completion results on NYU Depth v2 dataset [36]. We first upgrade S2D (c) to S2D-TR with efficient attention on 2D features, which significantly improves the initial depth completion accuracy (d). Based on the more accurate initial depths, DeCoTR uplifts 2D features to form a 3D point cloud and leverages cross-attention on 3D points, which leads to highly accurate depth completion, with sharp details and close-to-GT quality (e). We highlight sample regions where we can clearly see progressively improving depths by using our proposed designs.

former layers on bottleneck and skip connection features. This upgraded model, termed *S2D-TR*, achieves significantly improved depth accuracy, on par with state-of-the-art models, without requiring any iterative spatial propagation.

Given the initial depth map, we uplift 2D features to 3D to form a point cloud, which is subsequently processed by transformer layers, to which we refer as *3D-TR* layers. Prior to feeding the points to transformer layers, we normalize them, which regularizes the 3D feature learning and leads to better accuracy. In each 3D-TR layer, we follow standard practice [40, 46] to perform neighborhood-based attention, as global attention would be computationally intractable when the number of points is large. To facilitate long-range contextual understanding, we additionally incorporate global attention on lower-scale versions of the point cloud. Finally, 3D features are projected back to the 2D image plane and consumed by a decoder to produce the final depth prediction. As we shall see in the paper, our proposed transformer-based learning in full 3D provides considerably improved accuracy and generalizability for depth completion; see Fig. 1 for a visual example.

In summary, our main contributions are as follows:

- We present DeCoTR, a novel transformer-based approach to perform full 3D feature learning for depth completion. This enables high-quality depth estimation without requiring iterative processing steps.
- In order to properly do this, we upgrade the commonly used initial network S2D, by enhancing its bottleneck and skip connection features using transformers. The resulting model, S2D-TR, performs on-par with SOTA and provides more correct depths to subsequent 3D learning.
- We devise useful techniques to normalize the uplifted 3D feature point cloud, which improves the model learning. We additionally apply low-resolution global attention to 3D points, which enhances long-range understanding without making computation infeasible.

- Through extensive evaluations on standard benchmarks, NYU Depth v2 [36] and KITTI [14], we demonstrate the efficacy of DeCoTR and show that it sets the new SOTA, e.g., new best result on NYU Depth v2. Our zero-shot testing on ScanNet [7] and DDAD [15] further showcases the better generalizability of our model as compared to existing methods.

## 2. Related Works

**Depth completion:** Early depth completion approaches [16, 22, 38] rely solely on the sparse depth measurements to estimate the dense depth. Since these methods do not utilize the image, they usually suffer from artifacts like blurriness, especially at object boundaries. Later, image-guided depth completion alleviates these issues by incorporating the image. S2D [27], one of the first papers on this, leverages a convolutional network to consume both the image and sparse depth map. Subsequent papers design more sophisticated convolutional models for depth completion [3, 19, 31, 37, 48]. In order to enhance depth quality, researchers have studied various spatial propagation algorithms [4, 5, 24, 29]. These solutions utilize depth values and features given by an initial network (usually S2D), and performs iterative steps to mix and aggregate features on the 2D image plane. In many papers nowadays, it has become a common practice to use spatial propagation on top of the proposed depth completion network in order to achieve state-of-the-art accuracy [17, 28, 45]. Some recent works more tightly integrate iterative processing into the network, using architectures like recurrent network [39] and repetitive hourglass network [42].

While existing solutions predominately propose architectures to process features on 2D, several works explore 3D representations. For instance, [18, 44, 49] considers the sparse depth as a point cloud and learn features from it. However, the extremely sparse points present a challenge

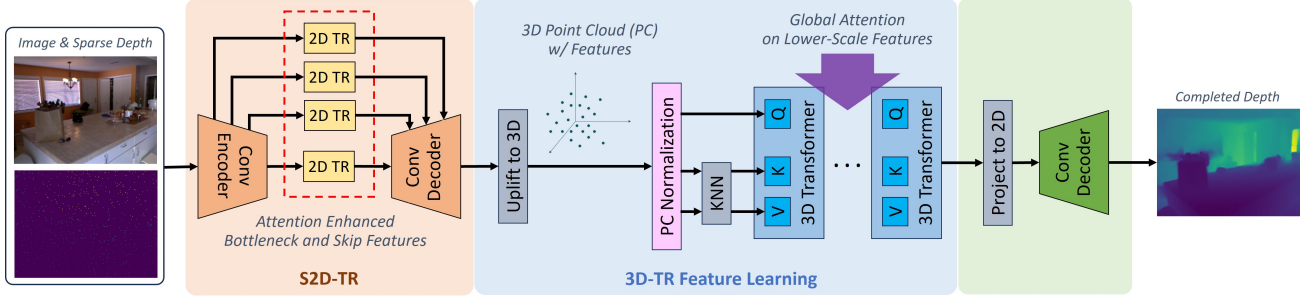


Figure 2. Overview of our proposed DeCoTR. The input RGB image and sparse depth map are first processed by our S2D-TR, which upgrades S2D with efficient 2D attentions. The learned 2D guidance features from S2D-TR are then uplifted to form a 3D feature point cloud based on the initial completed depth map. We normalize the point cloud and feed it through multiple 3D cross-attention layers (3D-TR) to enable geometry-aware feature learning and processing. We also introduce efficient global attention to capture long-range scene context. The attended 3D features from 3D-TR are projected back to 2D and given to a decoder to output the final completed depth map.

to feature learning. One of these works, GraphCSPN [26], employs S2D as an initial network to generate the full depth map, before creating a denser point cloud and performing feature learning on it. However, this is limited by the insufficient accuracy of the initial depths by S2D and still needs iterative processing to achieve good accuracy.

**Vision transformer:** Even since its introduction [10], vision transformers have been extensively studied and utilized for various computer vision tasks, including classification, detection, segmentation, depth estimation, tracking, 3D reconstruction, and more. We refer readers to these surveys for a more comprehensive coverage of these works. More related to our paper are those that leverage vision transformers for depth completion, such as CompletionFormer [45] and GuideFormer [32]. While they demonstrate the effectiveness of using vision transformers for depth completion, their feature learning is only performed on the 2D image plane. A very recent paper, PointDC [44], proposes to apply transformer to 3D point cloud in the depth completion pipeline. However, PointDC operates on very sparse points, which makes it challenging for learning 3D features.

### 3. Method

In this section, we present our proposed approach, DeCoTR, powered by efficient 2D and powerful 3D attention learning. The overall pipeline of DeCoTR is shown in Fig. 2.

#### 3.1. Problem Setup

Given aligned sparse depth map  $S \in \mathbb{R}^{H \times W}$  and an RGB image  $I \in \mathbb{R}^{H \times W \times 3}$ , the goal of image-guided depth completion is to recover a dense depth map  $D \in \mathbb{R}^{H \times W}$  based on  $S$  and with semantic guidance from  $I$ . The underlying reasoning is that visually similar adjacent regions are likely to have similar depth values. Formally, we have

$$D = H(S, I), \quad (1)$$

where  $H$  is a depth completion model to be learned.

It is a common approach to employ early fusion between the depth and RGB modalities [26, 27, 29]. This has the advantage of enabling features to contain both RGB and depth information early on, so that the model can learn to rectify incorrect depth values by leveraging neighboring, similar pixels that have correct depths. We follow the same practice, first encoding RGB  $I$  and sparse depth  $S$  with two separate convolutions to obtain image and depth features  $f_I$  and  $f_S$  respectively:

$$f_I = \text{conv}_{rgb}(I), \quad f_S = \text{conv}_{dep}(S) \quad (2)$$

which are then concatenated channel-wise to generate the initial fused feature,  $f_1 \in \mathbb{R}^{C_1 \times H_1 \times W_1}$ .

#### 3.2. Enhancing Baseline with Efficient 2D Attention

The early-fusion architecture of S2D [27] has been commonly used by researchers as a base network to predict an initial completed depth map (e.g., [26, 29]). Given the initial fused feature  $f_1$ , S2D continues to encode  $f_1$  and generates multi-scale features  $f_m \in \mathbb{R}^{C_m \times H_m \times W_m}$  for  $m = \{2, \dots, 5\}$ , where  $C_m$ ,  $H_m$ ,  $W_m$  are the number of channels, and height and width of the feature map. A conventional decoder with convolutional and upsampling layers is used to consume these features, where the smallest feature map is directly fed to the decoder and larger ones are fed via skip connections. The decoder has two prediction branches, one producing a completed depth map and the other generates guidance feature  $g$ .

This architecture, however, has limited accuracy and may provide erroneous depth values for subsequent operations, such as 2D spatial propagation or 3D representation learning. As such, we propose to leverage self-attentions to enhance the S2D features. More specifically, we apply Multi-Headed Self-Attention (MHSA) to each  $f_m$ . Since MHSA incurs quadratic complexity in both time and memory w.r.t to input resolution, in order to avoid the intractable costs of processing high-resolution feature maps, we first employ depth-separable convolutions [6] to summarize large feature maps and reduce their resolutions to

the same size (in terms of height, width, and channel) as the smallest feature map. The downsized feature maps are denoted as  $\tilde{f}_m \in \mathbb{R}^{C_k \times H_k \times W_k}$ ,  $\forall m > 1$ . Three linear layers are used to derive query  $\tilde{q}_i \in \mathbb{R}^{N_k \times C_k}$ , key  $\tilde{k}_m \in \mathbb{R}^{N_k \times C_k}$ , value  $\tilde{v}_m \in \mathbb{R}^{N_k \times C_k}$  for each  $\tilde{f}_m$ , where  $N_k = H_k \cdot W_k$ . Next, we apply self-attention for each  $\tilde{f}_m$ :

$$\tilde{f}_m^A = \text{softmax}\left(\frac{\tilde{q}_m \tilde{k}_m^\top}{\sqrt{C_k}}\right) \tilde{v}_m, \quad (3)$$

where  $\tilde{f}_m^A$  denotes the attended features. These features are restored to their original resolutions by using depth-separable de-convolutions and we denote the restored versions as  $f_m^A$ . Finally, we apply a residual addition to obtain the enhanced feature map for each scale:

$$f_m^E = f_m^A + f_m, \quad \forall m = \{2, \dots, 5\}. \quad (4)$$

The enhanced feature maps are then consumed by the decoder to predict the initial completed depth map and guidance feature map. Our upgraded version of S2D with efficient attention enhancement, denoted as S2D-TR, provides significantly improved accuracy while having better efficiency than latest transformer-based depth completion models. For instance, S2D-TR achieves a lower RMSE (0.094 vs. 0.099) with  $\sim 50\%$  less computation as compared to CompletionFormer without spatial propagation [45].

### 3.3. Feature Cross-Attention in 3D

Considering the 3D nature of depth completion, it is important for the model to properly exploit 3D geometric information when processing the features. To enable this, we first un-project 2D guidance feature from S2D-TR, based on the initial completed depth, to form a 3D point cloud. This is done as follows, assuming a pinhole camera model with known intrinsic parameters:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = d \begin{bmatrix} 1/\gamma_u & 0 & -c_u/\gamma_u & 0 \\ 0 & 1/\gamma_v & -c_v/\gamma_v & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \\ 1/d \end{bmatrix} \quad (5)$$

where  $\gamma_u, \gamma_v$  are focal lengths,  $(c_u, c_v)$  is the principal point,  $(u, v)$  and  $(x, y, z)$  are the 2D pixel coordinates and 3D coordinates, respectively.

Given the large number of 3D points uplifted from the 2D feature map, it is computationally intractable to perform attention on all the points simultaneously. As such, we adopt a neighborhood-based attention, by finding the K-Nearest-Neighboring (KNN) points for each point in the point cloud,  $p_i \in \mathbb{R}^3$ , which we denote as  $\mathcal{N}(i)$ .

To bake 3D geometric relationship into the feature learning process, we perform cross-attention between the feature of each point and features of its neighboring points. Concretely, we modify from the original point transformer [40, 46] to implement this. For each point  $p_i$ , linear projections are first applied to transform its guidance feature  $g_i$  to query

$q_i$ , key  $k_i$ , and value  $v_i$ . Following [46], we use vector attention which creates attention weights to modulate individual feature channels. More specifically, the 3D cross-attention is performed as follows:

$$a_{ij} = w(\phi(q_i, k_j)), \quad (6)$$

$$g_i^a = \sum_{j \in \mathcal{N}(i)} \text{softmax}(A_i)_j \odot v_j, \quad (7)$$

where  $\phi$  is a relation function to capture the similarity between a pair of input point features (we use subtraction here),  $w$  is a learnable encoding function that computes attention scores to re-weight the channels of the value,  $A$  is the attention weight matrix whose entries are  $a_{ij}$  for points  $p_i$  and  $p_j$ ,  $g_i^a$  denotes the output feature after cross-attention for  $p_i$ , and  $\odot$  denotes the Hadamard product. We perform such 3D cross-attention in multiple transformer layers, to which we refer as 3D-TR layers.

While it is possible to directly use existing point transformers off-the-shelf, we find that this is not optimal for depth completion. Specifically, we incorporate the following technical designs to improve the 3D feature learning process.

**Point cloud normalization:** We normalize the constructed point cloud from S2D-TR outputs into a unit ball, before proceeding to the 3D attention layers. We find this technique effectively improves depth completion, as we shall show in the experiments.

**Positional embedding:** Instead of the positional embedding multiplier proposed in [40], we adopt the conventional one based on relative position difference. We find the more complex positional embedding multiplier does not benefit the learning and incurs additional computational cost.

### 3.4. Capturing Global Context in 3D

The 3D cross-attention discussed previously updates each point feature only based on the point's estimated 3D neighborhood, in order to maintain computation tractability given the quadratic complexity of attention w.r.t. number of points. However, global or long-range scene context is also important for the model to develop accurate 3D understanding. To enable global understanding while keeping computation costs under control, we propose to perform global 3D cross-attention only on a downsampled point set, at the last encoding stage of the point transformer. In this case, we use the scalar attention as follows:

$$g_i^{ga} = \sum_{j \neq i} \text{softmax}\left(\frac{\langle q_i, k_j \rangle}{\sqrt{C_g}}\right) v_j, \quad (8)$$

where  $\langle \cdot \rangle$  denotes dot product and  $C_g$  is the embedding dimension. We apply global attention after the local neighborhood-based attentions.

### 3.5. Training

We train DeCoTR with a masked  $\ell_1$  loss between the final completed depth maps and the ground-truth depth maps,



following standard practice as in [26, 29]. More formally, the loss is given by

$$\mathcal{L}(D^{gt}, D^{pred}) = \frac{1}{N} \sum_{i,j} \mathbb{I}_{\{d_{i,j}^{gt} > 0\}} |d_{i,j}^{gt} - d_{i,j}^{pred}|, \quad (9)$$

where  $\mathbb{I}$  is the indicator function,  $d_{i,j}^{gt} \in D^{gt}$  and  $d_{i,j}^{pred} \in D^{pred}$  represent pixel-wise depths in ground-truth and predicted depth maps, respectively,  $N$  is the total number of valid pixels.

## 4. Experiments

We conduct extensive experiments to evaluate our proposed DeCoTR on standard depth completion benchmarks and compare with the latest state-of-the-art (SOTA) solutions. We further perform zero-shot evaluation to assess the model generalizability and carry out ablation studies to analyze different parts of our proposed approach.

### 4.1. Experimental Setup

**Datasets:** We perform standard depth completion evaluations on NYU Depth v2 (NYUD-v2) [36] and KITTI Depth Completion (KITTI-DC) [13, 14], and generalization tests on ScanNet-v2 [7] and DDAD [15]. These datasets cover a variety of indoor and outdoor scenes. We follow the sampling settings from existing works to create input sparse depth [26, 29].

NYUD-v2 provides RGB images and depth maps captured by a Kinect device from 464 different indoor scenes. We use the official split: 249 scenes for training and the remaining 215 for testing. Following the common practice [26, 29, 45], we sample  $\sim 50,000$  images from the training set and resize the image size from  $480 \times 640$  first to half and then to  $228 \times 304$  with center cropping. We use the official test set of 654 images for evaluation.

KITTI is a large real-world dataset in the autonomous driving domain, with over 90,000 paired RGB images and LiDAR depth measurements. There are two versions of KITTI dataset used for depth completion. One is from [27], which consists of 46,000 images from the training sequences for training and a random subset of 3,200 images from the test sequences for evaluation. The other one is KITTI Depth Completion (KITTI-DC) dataset, which provides 86,000 training, 6,900 validation, and 1,000 testing samples with corresponding raw LiDAR scans and reference images. We use KITTI-DC to train and test our model on the official splits.

ScanNet-v2 contains 1,513 room scans reconstructed from RGB-D frames. The dataset is divided into 1,201 scenes for training and 312 for validation, and provides an additional 100 scenes for testing. For sparse input depths, we sample point clouds from vertices of the reconstructed meshes. We use the 100 test scenes to evaluate depth completion performance, with 20 frames randomly selected per

scene. We remove samples where more than 10% of the ground-truth depth values are missing, resulting in 745 test frames across all 100 test scenes.

DDAD is an autonomous driving dataset collected in the U.S. and Japan using a synchronized 6-camera array, featuring long-range (up to 250m) and diverse urban driving scenarios. Following [15], we downsample the images from the original resolution of  $1216 \times 1936$  to  $384 \times 640$ . We use the official 3,950 validation samples for evaluation. Since after downsampling there’s only less than 5% valid ground truth depth, for our method and all the comparing we sample all the available valid depth points so that reasonable results are generated.

**Implementation Details:** We implement our proposed approach using PyTorch [30]. We use the Adam [21] optimizer with an initial learning rate of  $5 \times 10^{-4}$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and no weight decay. The batch size for NYUDv2 and KITTI-DC per GPU is set to 8 and 4, respectively. All experiments are conducted on 8 NVIDIA A100 GPUs.

**Evaluation:** We use standard metrics to evaluate depth completion performance [12], including Root Mean Squared Error (RMSE), Absolute Relative Error (Abs Rel),  $\delta < 1.25$ ,  $\delta < 1.25^2$ , and  $\delta < 1.25^3$ . On KITTI-DC test, we use the official metrics: RMSE, MAE, iRMSE and iMAE. We refer readers to the supplementary file for detailed mathematical definitions of these metrics. The depth values are evaluated with maximum distances of 80 meters and 200 meters for KITTI and DDAD, respectively, and 10 meters for NYUD-v2 and ScanNet.

### 4.2. Results on NYUD-v2 and KITTI

**On NYUD-v2:** Table 1 summarizes the quantitative evaluation results on NYUD-v2. Our proposed DeCoTR approach sets the new SOTA performance, with the lowest RMSE of 0.086 outperforming all existing solutions. When not using 3D global attention, DeCoTR already provides the best accuracy and global attention further improves it. Specifically, our DeCoTR considerably outperforms latest SOTA methods that also leverage 3D representation and/or transformers, such as GraphCSPN, PointDC, and CompletionFormer. Note that although PointDC uses both 3D representation and transformer, it only obtains slightly lower RMSE when comparing to methods that do not use 3D or transformer (e.g., CompletionFormer, GraphCSPN). This indicates that the PointDC approach is suboptimal, potentially due to the extremely sparse 3D points.

Fig. 3 provides sample qualitative results on NYUD-v2. We see that DeCoTR generates highly accurate dense depth maps that are very close to the ground truth. The depth maps produced by DeCoTR capture much finer details as compared to existing SOTA methods. For instance, in the second example, our proposed approach accurately predicts

Method	RMSE ↓	Abs Rel ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
S2D [27]	0.204	0.043	97.8	99.6	99.9
DeepLiDAR [31]	0.115	0.022	99.3	99.9	100.0
CSPN [4]	0.117	0.016	99.2	99.9	100.0
DepthNormal [41]	0.112	0.018	99.5	99.9	100.0
ACMNet [47]	0.105	0.015	99.4	99.9	100.0
GuideNet [37]	0.101	0.015	99.5	99.9	100.0
TWISE [19]	0.097	<u>0.013</u>	99.6	99.9	100.0
NLSPN [29]	0.092	<b>0.012</b>	99.6	99.9	100.0
RigNet [42]	0.090	<u>0.013</u>	99.6	99.9	100.0
DySPN [24]	0.090	<b>0.012</b>	99.6	99.9	100.0
CompletionFormer [45]	0.090	<b>0.012</b>	-	-	-
PRNet [23]	0.104	0.014	99.4	99.9	100.0
CostDCNet [20]	0.096	<u>0.013</u>	99.5	99.9	100.0
PointFusion [18]	0.090	0.014	99.6	99.9	100.0
GraphCSPN [26]	0.090	<b>0.012</b>	99.6	99.9	100.0
PointDC [44]	0.089	<b>0.012</b>	99.6	99.9	100.0
DeCoTR (ours)	<u>0.087</u>	<b>0.012</b>	99.6	99.9	100.0
DeCoTR w/ GA (ours)	<b>0.086</b>	<b>0.012</b>	99.6	99.9	100.0

Table 1. Quantitative evaluation of depth completion performance on NYU-Depth-v2. GA denotes global attention. RMSE and REL are in meters. Methods in the top part of the table focus on feature learning and processing in 2D and those in the bottom block exploit 3D representation. Best and second best numbers are highlighted in bold and underlined, respectively, for RMSE and Abs Rel.

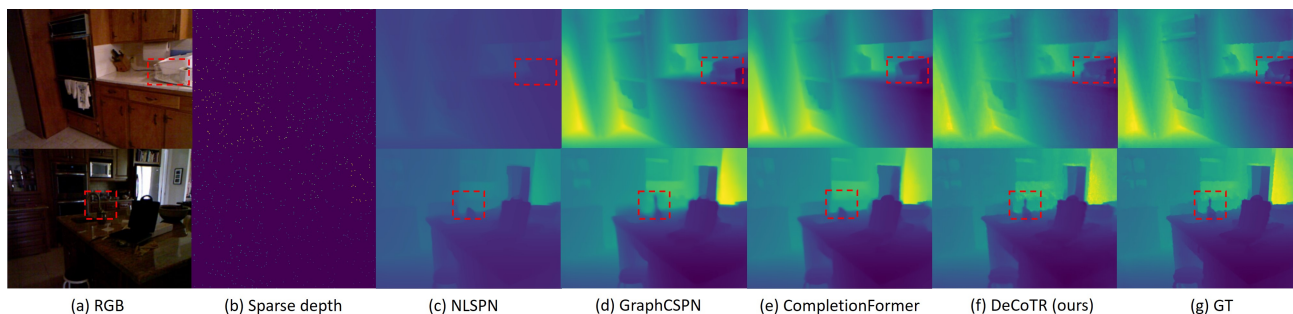


Figure 3. Qualitative results on NYUD-v2. We compare with SOTA methods such as NLSPN, GraphCSPN, and CompletionFormer. Areas where DeCoTR provides better depth accuracy are highlighted.

the depth on the faucet despite its small size in the images and the low contrast, while other methods struggle.

**On KITTI-DC:** We evaluate DeCoTR and compare with existing methods (including latest SOTA) on the official KITTI test set, as shown in Table 2. DeCoTR achieves SOTA depth completion accuracy and is among the top-ranking methods on KITTI-DC leaderboard.<sup>1</sup> We see that DeCoTR performs significantly better than existing SOTA methods that leverage 3D representations, e.g., GraphCSPN, PointDC. This indicates that DeCoTR has the right combination of dense 3D representation and transformer-based learning.

Fig. 4 shows visual examples of our completed depth maps on KITTI. DeCoTR is able to generate correct depth prediction where NLSPN produces erroneous depth values;

<sup>1</sup>Top-5 among published methods at the time of submission, in terms of iRMSE, iMAE, and MAE.

see the highlighted areas in the figure. For instance, in the second example, DeCoTR accurately estimates the depth around the upper edge of the truck while the depth map by NLSPN is blurry in that region.

### 4.3. Zero-Shot Testing on ScanNet and DDAD

Most existing papers only evaluate their models on NYUD-v2 and KITTI, without looking into model generalizability. In this part, we perform cross-dataset evaluation. More specifically, we run zero-shot testing of NYUD-v2-trained models on ScanNet-v2 and KITTI-trained models on DDAD. This will allow us to understand how well our DeCoTR as well as existing SOTA models generalize to data not seen in training.

Tables 3 and 4 present evaluation results on ScanNet-v2 and DDAD, respectively. We see that DeCoTR generalizes better to unseen datasets when comparing to existing SOTA models. It is noteworthy to mention that on DDAD,

Method	RMSE ↓	MAE ↓	iRMSE ↓	iMAE ↓
CSPN [4]	1019.64	279.46	2.93	1.15
TWISE [19]	840.20	195.58	2.08	<b>0.82</b>
ACMNet [47]	744.91	206.09	2.08	0.90
GuideNet [37]	736.24	218.83	2.25	0.99
NLSPN [29]	741.68	199.59	1.99	<u>0.84</u>
PENet [17]	730.08	210.55	2.17	0.94
GuideFormer [32]	721.48	207.76	2.14	0.97
RigNet [42]	712.66	203.25	2.08	0.90
DySPN [24]	<u>709.12</u>	<b>192.71</b>	<b>1.88</b>	<b>0.82</b>
CompletionFormer [45]	<b>708.87</b>	203.45	2.01	0.88
PRNet [23]	867.12	204.68	2.17	0.85
FuseNet [3]	752.88	221.19	2.34	1.14
PointFusion [18]	741.9	201.10	1.97	0.85
GraphCSPN [26]	738.41	199.31	1.96	<u>0.84</u>
PointDC [44]	736.07	201.87	1.97	0.87
DeCoTR (ours)	717.07	<u>195.30</u>	<u>1.92</u>	<u>0.84</u>

Table 2. Quantitative evaluation of depth completion performance on official KITTI Depth Completion test set. RMSE and MAE are in millimeters, and iRMSE and iMAE are in 1/km. Similar to Table 1, methods in the top part focus on feature learning in 2D and those in the bottom block exploit 3D representation. Best and second best numbers are highlighted in bold and underlined, respectively.

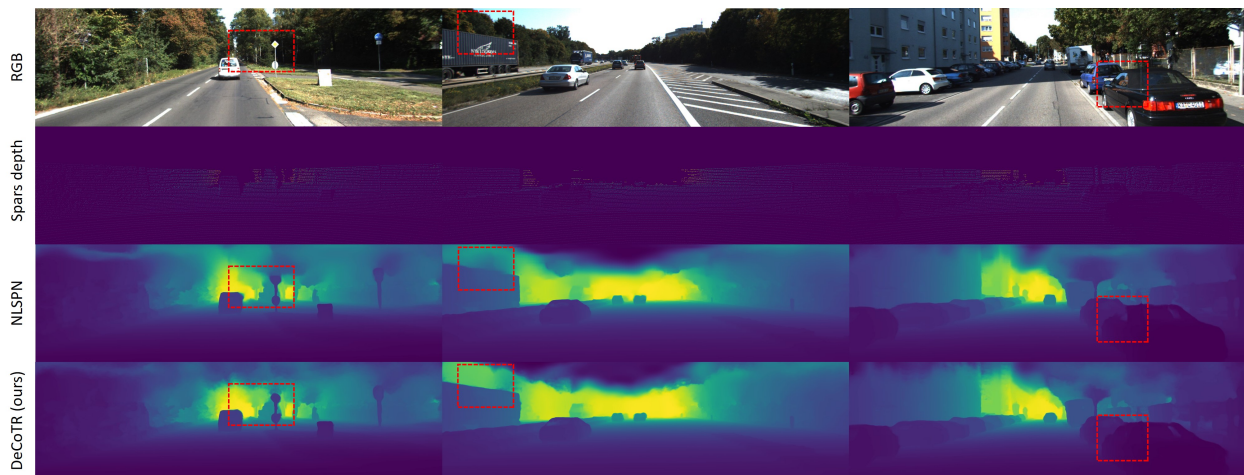


Figure 4. Qualitative results on KITTI DC. Areas where DeCoTR provides better depth accuracy are highlighted.

DeCoTR has significantly lower depth errors as compared to both NLSPN and CompletionFormer, despite that CompletionFormer has slightly lower RMSE on KITTI-DC test. Moreover, in this case, CompletionFormer has even worse accuracy than NLSPN, indicating its poor generalizability.

Fig. 5 shows sample visual results of zero-shot depth completion on ScanNet-v2. DeCoTR generates highly accurate depth maps and captures fine details, e.g., arm rest in the first example, lamp in the second example. Other methods cannot recover the depths accurately. Fig. 6 provides qualitative results on DDAD for CompletionFormer and our DeCoTR. While this is a challenging test setting given the much larger depth range in DDAD, DeCoTR still predicts reasonable depths. In contrast, it can be seen that CompletionFormer performs very poorly on DDAD. We notice that DeCoTR’s predictions are more accurate in the nearer

Method	RMSE ↓	$\delta < 1.25$ ↑
NLSPN [29]	0.198	97.3
GraphCSPN [26]	0.197	97.3
CompletionFormer [45]	0.194	97.3
DeCoTR (ours)	<b>0.188</b>	<b>97.6</b>

Table 3. Zero-shot testing on ScanNet-v2 using models trained on NYUD-v2. Best numbers are highlighted in bold.

range (e.g., on cars) and less so when it is far away (e.g., on trees), since KITTI training only covers up to 80 meters whereas DDAD has depth up to 200 meters. This is also confirmed by the lower-than-KITTI RMSE and higher-than-KITTI MAE numbers of DeCoTR on DDAD.



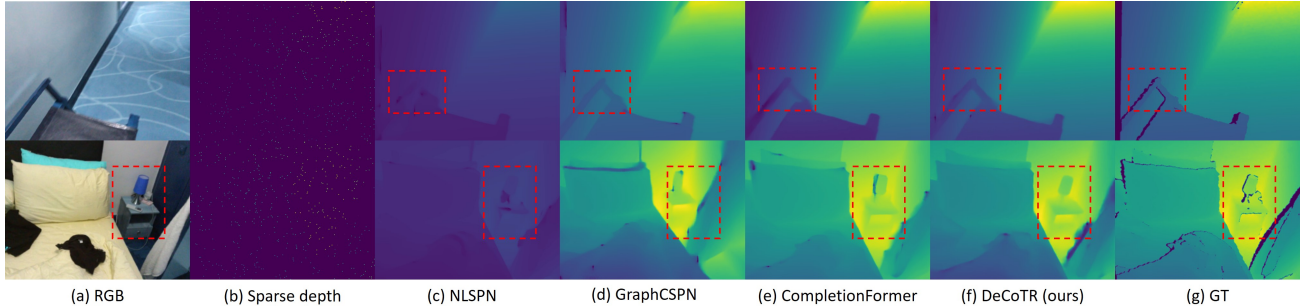


Figure 5. Qualitative results of zero-shot inference on ScanNet-v2. Areas where DeCoTR provide better depth accuracy are highlighted.

Method	RMSE ↓	MAE ↓
NLSPN [29]	701.9	309.6
CompletionFormer [45]	889.3	400.1
DeCoTR (ours)	<b>399.2</b>	<b>263.1</b>

Table 4. Zero-shot testing on DDAD using models trained on KITTI. Best numbers are highlighted in bold.

#### 4.4. Ablation Study

In this part, we investigate the effectiveness of various design aspects of our proposed DeCoTR solution. Table 5 summarizes the ablation study results. Starting from the S2D baseline, we significantly improve depth completion performance by introducing efficient attention on the 2D features, reducing RMSE from 0.204 to 0.094. Next, by using neighborhood-based cross-attention on the 3D points (without normalizing the point cloud before 3D-TR layers), we reduce RMSE to 0.089. Even though scaling a 3D scene to a uniform perceived range may present a challenge to maintain the original spatial relationship, after applying our normalization scheme, DeCoTR achieves a better RMSE of 0.087 and by additionally incorporating efficient global attention, the RMSE is further improved to 0.086. This study verifies the usefulness of our proposed components and techniques.

Note that if we directly apply 3D-TR on top of the original S2D network (second row in the table), we can still drastically improve upon S2D but fail to outperform existing methods that leverage 3D or transformers such as GraphCSPN and CompletionFormer. This confirms the importance of getting more accurate initial depth before applying 3D feature learning.

## 5. Conclusion

In this paper, we proposed a novel approach, DeCoTR, for image-guided depth completion, by employing transformer-based learning in full 3D. We first proposed an efficient attention scheme to upgrade the common baseline of S2D, allowing S2D-TR to provide more accurate initial depth completion. 2D features are then uplifted to form a 3D point cloud followed by 3D-TR layers that apply power-

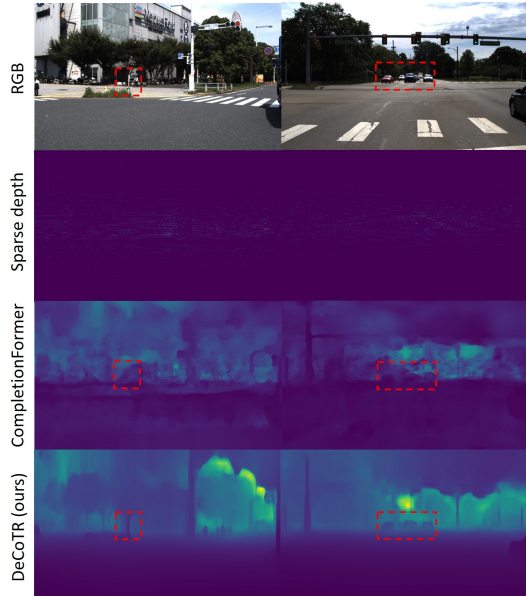


Figure 6. Qualitative results of zero-shot inference on DDAD. Areas where DeCoTR provide better depth accuracy are highlighted.

Method	RMSE ↓	$\delta < 1.25$ ↑
S2D [27]	0.204	97.8
S2D-TR	0.094	99.4
S2D + 3D-TR	0.091	99.6
DeCoTR w/o normalization	0.089	99.6
DeCoTR	0.087	99.6
DeCoTR w/ global attention	0.086	99.6

Table 5. Ablation study conducted on NYUD-v2.

ful neighborhood-based cross-attention on the 3D points. We further devised an efficient global attention operation to provide scene-level understanding while keeping computation costs in check. Through extensive experiments, we have shown that DeCoTR achieves SOTA performance on standard benchmarks like NYUD-v2 and KITTI-DC. Furthermore, zero-shot evaluation on unseen datasets such as ScanNet and DDAD shows that DeCoTR has better generalizability as compared to existing methods.



## References

- [1] Lin Bai, Yiming Zhao, Mahdi Elhousni, and Xinming Huang. Depthnet: Real-time lidar point cloud depth completion for autonomous vehicles. *IEEE Access*, 8:227825–227833, 2020. 1
- [2] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE international conference on computer vision*, pages 2722–2730, 2015. 1
- [3] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10023–10032, 2019. 1, 2, 7
- [4] 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. 1, 2, 6, 7
- [5] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2361–2379, 2019. 1, 2
- [6] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 3
- [7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2, 5
- [8] Jorge de Heuvel, Nathan Corral, Benedikt Kreis, Jacobus Conradi, Anne Driemel, and Maren Bennewitz. Learning depth vision-based personalized robot navigation from dynamic demonstrations in virtual reality. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6757–6764. IEEE, 2023. 1
- [9] Catherine Diaz, Michael Walker, Danielle Albers Szafir, and Daniel Szafir. Designing for depth perceptions in augmented reality. In *2017 IEEE international symposium on mixed and augmented reality (ISMAR)*, pages 111–122. IEEE, 2017. 1
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 3
- [11] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, et al. Depthlab: Real-time 3d interaction with depth maps for mobile augmented reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 829–843, 2020. 1
- [12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 5
- [13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 5
- [14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 5
- [15] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2485–2494, 2020. 2, 5
- [16] Simon Hawe, Martin Kleinsteuber, and Klaus Diepold. Dense disparity maps from sparse disparity measurements. In *2011 International Conference on Computer Vision*, pages 2126–2133. IEEE, 2011. 2
- [17] 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. IEEE, 2021. 1, 2, 7
- [18] Lam Huynh, Phong Nguyen, Jiří Matas, Esa Rahtu, and Janne Heikkilä. Boosting monocular depth estimation with lightweight 3d point fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12767–12776, 2021. 1, 2, 6, 7
- [19] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2583–2592, 2021. 2, 6, 7
- [20] Jaewon Kam, Jungeon Kim, Soongjin Kim, Jaesik Park, and Seungyong Lee. Costdcnet: Cost volume based depth completion for a single rgb-d image. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, page 257–274, Berlin, Heidelberg, 2022. Springer-Verlag. 6
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5
- [22] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 16–22. IEEE, 2018. 2
- [23] Byeong-Uk Lee, Kyunghyun Lee, and In So Kweon. Depth completion using plane-residual representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13916–13925, 2021. 6, 7
- [24] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1638–1646, 2022. 1, 2, 6, 7
- [25] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. *Advances in Neural Information Processing Systems*, 30, 2017. 1

- [26] Xin Liu, Xiaofei Shao, Bo Wang, Yali Li, and Shengjin Wang. Graphcspn: Geometry-aware depth completion via dynamic gcns. In *European Conference on Computer Vision*, pages 90–107. Springer, 2022. 1, 3, 5, 6, 7
- [27] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4803. IEEE, 2018. 1, 2, 3, 5, 6, 8
- [28] Danish Nazir, Alain Pagani, Marcus Liwicki, Didier Stricker, and Muhammad Zeshan Afzal. Semattnet: Toward attention-based semantic aware guided depth completion. *IEEE Access*, 10:120781–120791, 2022. 2
- [29] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 120–136. Springer, 2020. 1, 2, 3, 5, 6, 7, 8
- [30] 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. *Advances in neural information processing systems*, 32, 2019. 5
- [31] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. DeepLidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3313–3322, 2019. 2, 6
- [32] Kyeongha Rho, Jinsung Ha, and Youngjung Kim. Guideformer: Transformers for image guided depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6250–6259, 2022. 1, 3, 7
- [33] Yunxiao Shi, Haoyu Fang, Jing Zhu, and Yi Fang. Pairwise attention encoding for point cloud feature learning. In *2019 International Conference on 3D Vision (3DV)*, pages 135–144. IEEE, 2019. 1
- [34] Yunxiao Shi, Jing Zhu, Yi Fang, Kuochin Lien, and Junli Gu. Self-supervised learning of depth and ego-motion with differentiable bundle adjustment. *arXiv preprint arXiv:1909.13163*, 2019.
- [35] Yunxiao Shi, Hong Cai, Amin Ansari, and Fatih Porikli. Ega-depth: Efficient guided attention for self-supervised multi-camera depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 119–129, 2023. 1
- [36] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012. 2, 5
- [37] 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. 1, 2, 6, 7
- [38] 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. IEEE, 2017. 2
- [39] Yufei Wang, Bo Li, Ge Zhang, Qi Liu, Tao Gao, and Yuchao Dai. Lrru: Long-short range recurrent updating networks for depth completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9422–9432, 2023. 2
- [40] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 2, 4
- [41] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints, 2019. 1, 6
- [42] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. In *European Conference on Computer Vision*, pages 214–230, 2022. 2, 6, 7
- [43] Rajeev Yasarla, Hong Cai, Jisoo Jeong, Yunxiao Shi, Rishkek Garrepalli, and Fatih Porikli. Mamo: Leveraging memory and attention for monocular video depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8754–8764, 2023. 1
- [44] Zhu Yu, Zehua Sheng, Zili Zhou, Lun Luo, Si-Yuan Cao, Hong Gu, Huaqi Zhang, and Hui-Liang Shen. Aggregating feature point cloud for depth completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8732–8743, 2023. 1, 2, 3, 6, 7
- [45] Youmin Zhang, Xianda Guo, Matteo Poggi, Zheng Zhu, Guan Huang, and Stefano Mattoccia. Completionformer: Depth completion with convolutions and vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18527–18536, 2023. 1, 2, 3, 4, 5, 6, 7, 8
- [46] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2, 4
- [47] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing*, 30:5264–5276, 2021. 6, 7
- [48] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing*, 30:5264–5276, 2021. 2
- [49] Wending Zhou, Xu Yan, Yinghong Liao, Yuankai Lin, Jin Huang, Gangming Zhao, Shuguang Cui, and Zhen Li. Bev@dc: Bird’s-eye view assisted training for depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2023. 1, 2
- [50] Jing Zhu, Yunxiao Shi, Mengwei Ren, Yi Fang, Kuochin Lien, and Junli Gu. Structure-attended memory

network for monocular depth estimation. *arXiv preprint arXiv:1909.04594*, 2019. 1

- [51] Jing Zhu, Yunxiao Shi, Mengwei Ren, and Yi Fang. Mdanet: memorable domain adaptation network for monocular depth estimation. In *British Machine Vision Conference 2020*, 2020. 1