This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

# CompletionFormer: Depth Completion with Convolutions and Vision Transformers

Youmin Zhang[1] Xianda Guo[2] Matteo Poggi[1]
Zheng Zhu[2] Guan Huang[2] Stefano Mattoccia[1]

[1] University of Bologna [2] PhiGent Robotics

[1]{youmin.zhang2, m.poggi, stefano.mattoccia}@unibo.it

## Abstract

*Given sparse depths and the corresponding RGB images, depth completion aims at spatially propagating the sparse measurements throughout the whole image to get a dense depth prediction. Despite the tremendous progress of deep-learning-based depth completion methods, the locality of the convolutional layer or graph model makes it hard for the network to model the long-range relationship between pixels. While recent fully Transformer-based architecture has reported encouraging results with the global receptive field, the performance and efficiency gaps to the well-developed CNN models still exist because of its deteriorative local feature details. This paper proposes a Joint Convolutional Attention and Transformer block (JCAT), which deeply couples the convolutional attention layer and Vision Transformer into one block, as the basic unit to construct our depth completion model in a pyramidal structure. This hybrid architecture naturally benefits both the local connectivity of convolutions and the global context of the Transformer in one single model. As a result, our CompletionFormer outperforms state-of-the-art CNNs-based methods on the outdoor KITTI Depth Completion benchmark and indoor NYUv2 dataset, achieving significantly higher efficiency (nearly 1/3 FLOPs) compared to pure Transformer-based methods. Code is available at* `https://github.com/youmi-zym/CompletionFormer`.

## 1. Introduction

Active depth sensing has achieved significant gains in performance and demonstrated its utility in numerous applications, such as autonomous driving and augmented reality. Although depth maps captured by existing commercial depth sensors (*e.g.*, Microsoft Kinect [23], Intel RealSense [11]) or depths points within the same scanning line of LiDAR sensors are dense, the distance between valid/correct depth points could still be far owing to the sensor noise, challenging conditions such as transparent, shining, and dark surfaces, or the limited number of scanning lines of LiDAR sensors. To address these issues, depth completion [2, 16, 26, 31], which targets at completing and reconstructing the whole depth map from sparse depth measurements and a corresponding RGB image (*i.e.*, RGBD), has gained much attention in the latest years.

For depth completion, one key point is to get the depth affinity among neighboring pixels so that reliable depth labels can be propagated to the surroundings [2, 3, 8, 16, 26]. Based on the fact that the given sparse depth could be highly sparse due to noise or even no measurement being returned from the depth sensor, it requires depth completion methods to be capable of 1) detecting depth outliers by measuring the spatial relationship between pixels in both local and global perspectives; 2) fusing valid depth values from close or even extremely far distance points. All these properties ask the network for the potential to capture both local and global correlations between pixels. Current depth completion networks collect context information with the widely used convolution neural networks (CNNs) [2, 3, 8, 16, 26, 29, 37, 51] or graph neural network [42, 49]. However, both the convolutional layer and graph models can only aggregate within a local region, *e.g.* square kernel in 3×3 for convolution and kNN-based neighborhood for graph models [42, 49], making it still tough to model global long-range relationship, in particular within the shallowest layers of the architecture. Recently, GuideFormer [31] resorts fully Transformer-based architecture to enable global reasoning. Unfortunately, since Vision Transformers project image patches into vectors through a single step, this causes the loss of local details, resulting in ignoring local feature details in dense prediction tasks [28, 43]. For depth completion, the limitations affecting pure CNNs or Transformer based networks also manifest, as shown in Fig. 1. Despite *any* distance the reliable depth points could be distributed at, exploring an elegant integration of these two distinct paradigms, *i.e.* CNNs and Transformer, has not

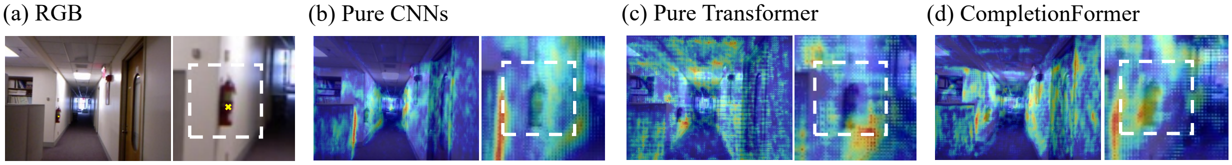


Figure 1. **Comparison of attention maps of pure CNNs, Vision Transformer, and the proposed CompletionFormer with joint CNNs and Transformer structure.** The pixel highlighted with a yellow cross in RGB image (a) is the one we want to observe how the network predicts it. Pure CNNs architecture (b) activates discriminative local regions (*i.e*., the region on the fire extinguisher), whereas pure Transformer based models (c) activate globally yet fail on local details. In contrast, our full CompletionFormer (d) can retain both the local details and global context.

been studied for depth completion yet.

In this work, we propose CompletionFormer, a pyramidal architecture coupling CNN-based local features with Transformer-based global representations for enhanced depth completion. Generally, there are two gaps we are facing: 1) the content gap between RGB and depth input; 2) the semantic gap between convolution and Transformer. As for the multimodal input, we propose embedding the RGB and depth information at the early network stage. Thus our CompletionFormer can be implemented in an efficient single-branch architecture as shown in Fig. 2 and multimodal information can be aggregated throughout the whole network. Considering the integration of convolution and Transformer, previous work has explored from several different perspectives [6,12,25,28,43] on image classification and object detection. Although state-of-the-art performance has been achieved on those tasks, high computation cost [12] or inferior performance [6, 12] are observed when these networks are directly adapted to depth completion task. To promise the combination of self-attention and convolution still being efficient, and also effective, we embrace convolutional attention and Transformer into one block and use it as the basic unit to construct our network in a multi-scale style. Specifically, the Transformer layer is inspired by Pyramid Vision Transformer [39], which adopts spatial-reduction attention to make the Transformer layer much more lightweight. As for the convolution-related part, the common option is to use plain convolutions such as Inverted Residual Block [32]. However, the huge semantic gap between convolution and the Transformer and the lost local details by Transformer require the convolutional layers to increase its own capacity to compensate for it. Following this rationale, we further introduce spatial and channel attention [40] to enhance convolutions. As a result, without any extra module to bridge the content and semantic gaps [12, 28, 31], every convolution and Transformer layer in the proposed block can access the local and global features. Hence, information exchange and fusion happen effectively at every block of our network.

To summarize, our main contributions are as follows:

- We propose integrating Vision Transformer with convolutional attention layers into one block for depth completion, enabling the network to possess both local and global receptive fields for multi-modal information interaction and fusion. In particular, spatial and channel attention are introduced to increase the capacity of convolutional layers.
- Taking the proposed Joint Convolutional Attention and Transformer (JCAT) block as the basic unit, we introduce a single-branch network structure, *i.e*. CompletionFormer. This elegant design leads to a comparable computation cost to current CNN-based methods while presenting significantly higher efficiency when compared with pure Transformer based methods.
- Our CompletionFormer yields substantial improvements to depth completion compared to state-of-the-art methods, especially when the provided depth is *very* sparse, as often occurs in practical applications.

## 2. Related Work

**Depth Completion.** Scene depth completion has become a fundamental task in computer vision with the emergence of active depth sensors. Recently, following the advance of deep learning, fully-convolutional network has been the prototype architecture for current state-of-the-art on depth completion. Ma *et al.* [21, 22] utilize a ResNet [7] based encoder-decoder architecture, *i.e.* U-Net, within either a supervised or self-supervised framework to predict the dense output. To preserve the accurate measurements in the given sparse depth and also perform refinement over the final depth map, CSPN [3] appends a convolutional spatial propagation network (SPN [18]) at the end of U-Net to refine its coarse prediction. Based on CSPN, learnable convolutional kernel sizes and a number of iterations are proposed to improve the efficiency [2], and the performance could be further improved by using unfixed local neighbors [26, 44] and independent affinity matrix for each iteration [16]. For all these SPN-based methods, while a larger context is observed within recurrent processing, the performance is limited by the capacity of the convolutional U-Net backbone.

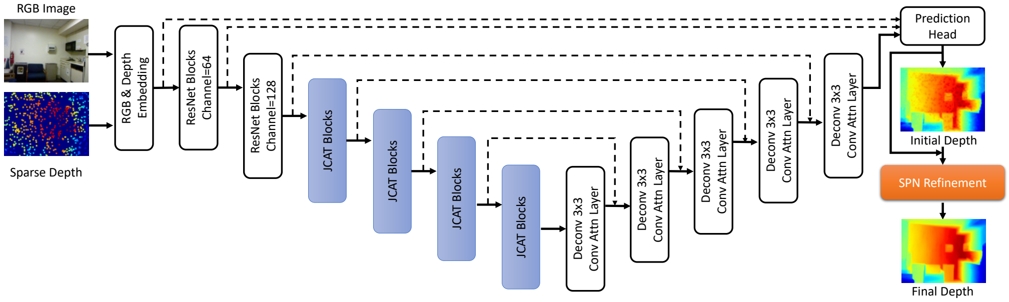


Figure 2. **CompletionFormer Architecture.** Given the sparse depth and corresponding RGB image, a U-Net backbone strengthened with JCAT block is used to perform the depth and image information interaction at multiple scales. Features from different stages are fused at full resolution and fed for initial prediction. Finally, a spatial propagation network (SPN) is exploited for final refinement.

Accordingly, we strengthen the expressivity of the U-Net backbone with local and global coherent context information, proving effective in improving performance.

Rather than depending on a single branch, multi-branch networks [8,17,24,29,35,37,46] are also adopted to perform multi-modal fusion. The common way to fuse the multi-modal information is simple concatenation or element-wise summation operation. More sophisticated strategies like image-guided spatially-variant convolution [35, 45], channel-wise canonical correlation analysis [50], neighbour attention mechanism [47] and attention-based graph propagation [42,49] were also proposed to enhance local information interaction and fusion. Instead of pixel-wise operation or local fusion, recently, GuideFormer [31] proposed a dual-branch fully Transformer-based network to embed the RGB and depth input separately, and an extra module is further designed to capture inter-modal dependencies. The independent design for each input source leads to huge computation costs (near 2T FLOPs with the $352 \times 1216$ input). In contrast, our CompletionFormer in one branch brings significant efficiency (559.5G FLOPs), and the included convolutional attention layer complements the disadvantage of a Transformer in local details.

**Vision Transformer.** Transformers [12, 19] are first introduced in natural language processing [38], then also showing great potential in the fields of image classification [4], object detection [12, 19, 43] and semantic segmentation [41]. Tasks related to 3D vision have also benefited from the enriched modeling capability of Transformer, such as stereo matching [13,15], supervised [14,30] and unsupervised monocular depth estimation [48], optical flow [10,34] and also depth completion [31]. Instead of relying on pure Vision Transformer [31], in this paper, we explore the combination of Transformer and convolution into one block for depth completion. Compared to the general backbone networks (*e.g*. ResNet [7] with fully CNN-based design, Swin Transformer [19] and PVT [39] based on pure Transformer, MPViT [12] and CMT [6] using both the convolutions and Vision Transformer), our proposed joint convolutional attention and Transformer block achieves much higher efficiency and performance on public benchmarks [33,36]

# 3. Method

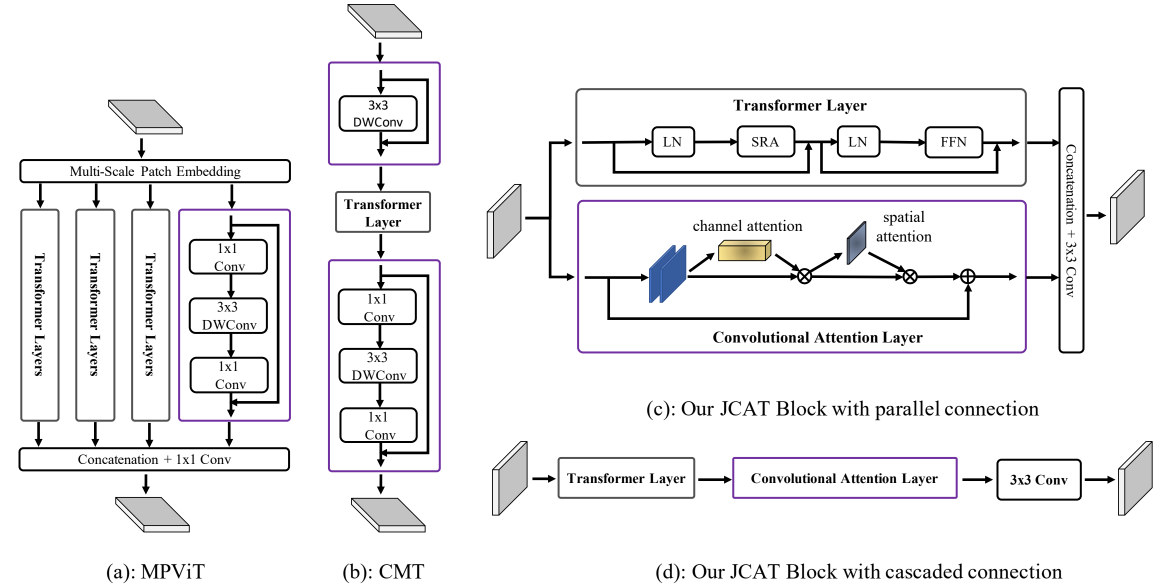
In practical applications, depth maps captured by sensors present various levels of sparsity and noise. Our goal is to introduce both the local features and global context information into the depth completion task so that it can gather reliable depth hints from any distance. The overall diagram of our CompletionFormer is shown in Fig. 2. After obtaining depth and RGB image embedding, a backbone constructed by our JCAT block is used for feature extraction at multiple scales and the decoder provides full-resolution features for initial depth prediction. Finally, for the purpose of preserving accurate depth from the sparse input, we refine the initial estimation with a spatial propagation network.

## 3.1. RGB and Depth Embedding

For depth completion, multimodal information fusion at an early stage has several advantages, 1) it makes the feature vector of each pixel possess both the RGB and depth information so that pixels with invalid depth still have a chance to be corrected by reliable depth measurements according to appearance similarity; 2) only one branch is required for the following network, which enables much efficient implementation. Therefore, we firstly use two separate convolutions to encode the input sparse depth map $S$ and RGB image $I$. The outputs are concatenated and further processed by another convolution layer to get the raw feature containing contents from both sources.

## 3.2. Joint Convolutional Attention and Transformer Encoder

It has been extensively studied how to build connections between pixels to implement depth propagation from reliable pixels while avoiding incorrect ones. Recently, convolution layer [2, 3, 8, 16, 26, 29, 37, 51] or attention based



(c): Our JCAT Block with parallel connection

(a): MPViT

(b): CMT

(d): Our JCAT Block with cascaded connection

Figure 3. **Example of architecture with convolutions and Vision Transformer.** (a) Multi-Path Transformer Block of MPViT [12]. (b) CMT Block of CMT-S [6]. (c) Our proposed JCAT block which contains two parallel streams, *i.e*. convolutional attention and Transformer layer respectively. (d) The variant of our proposed block with cascaded connection.

graph propagation [42, 49] has been the dominant operation for this purpose. Although a fully Transformer-based network [31] has also been adopted for this purpose, it shows worse results and much higher computational cost compared to pure CNNs-based methods. Considering the complementary properties of these two-style operations, an elegant integration of these two paradigms is highly demanded for depth completion task. On the other hand, for classification and object detection tasks, MPViT [12] and CMT [6] are two representative state-of-the-art networks on the combination of self-attention and convolution as shown in Fig. 3 (a) and (b) respectively. Generally, the integration can be implemented in a parallel or cascaded manner. Accordingly, inspired by their design, within CompletionFormer, we propose a joint design as shown in Fig. 3 (c) and (d). To decrease computation overhead but also get highly accurate depth completion result, our CompletionFormer only contains single rather than multiple time-consuming Transformer-based paths as in MPViT [12]. Furthermore, the representation power of convolution-based path is enhanced with spatial and channel attention.

Specifically, our encoder has five stages, allowing features representation at different scales to communicate with each other effectively. In the first stage, to decrease the computation cost and memory overhead introduced by the Transformer layer, we use a series of BasicBlocks from ResNet34 [7] to process and finally get downsampled feature map $F_1$ at half resolution. For the next four stages, we introduce our proposed JCAT block as the basic unit for framework design.

| CompletionFormer | #Layers | Params (M) | FLOPs (G) |
|---|---|---|---|
| Tiny | [2, 2, 2, 2] | 41.5 | 191.7 |
| Small | [3, 3, 6, 3] | 78.3 | 231.8 |
| Base | [3, 3, 18, 3] | 142.4 | 301.9 |

Table 1. **CompletionFormer Configurations.** #Layers denotes the number of our JCAT blocks in each stage. For all model variants, the channels of 4 stages are 64, 128, 320, 512, respectively. FLOPs are measured using 480 × 640 input image.

Basically, for each stage $i \in \{2, 3, 4, 5\}$, it consists of a patch embedding module and $L_i$ repeated JCAT blocks. The patch embedding module firstly divides the feature map $F_{i-1}$ from previous stage $i - 1$ into patches with size $2 \times 2$. We implement it with a $3 \times 3$ convolution layer and stride set to 2 as [39], so it actually halves resolution for features $F_{i-1}$ and thus allows for obtaining a features pyramid $\{F_2, F_3, F_4, F_5\}$, whose resolutions are $\{1/4, 1/8, 1/16, 1/32\}$ with respect to the input image. Furthermore, position embedding is also included in the embedded patches and passed through the JCAT blocks.

**Joint Convolutional Attention and Transformer Block.** In overview, our JCAT block can be organized in a parallel or cascaded manner as shown in Fig. 3 (c) and (d) respectively. The Transformer layer is implemented in an efficient way as in Pyramid Vision Transformer [39], which contains a spatial-reduction attention (SRA) layer with multi-head mechanism and a feed-forward layer (FNN).

Given input features $F \in \mathbb{R}^{H_i \times W_i \times C}$ from the patch embedding module or last joint block (with $H_i$ and $W_i$ height and width of features at stage $i$, and $C$ number of channel), we firstly normalize it with layer normalization [1] (LN) and then flatten it into vector tokens $X \in \mathbb{R}^{N \times C}$, where $N$ is the number of tokens and equals to $H_i \times W_i$, *i.e.* the number of all pixels in the $F$. Using learned linear transformations $W^Q$, $W^K$, and $W^V \in \mathbb{R}^{C \times C}$, tokens $X$ are projected into corresponding query $Q$, key $K$, and value vectors $V \in \mathbb{R}^{N \times C}$. Here, the spatial scale of $K$ and $V$ is further reduced to decrease memory consumption, and then self-attention is performed as:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{C_{head}}})V, \qquad (1)$$

with $C_{head}$ the channel dimension of each attention head in SRA. According to Eq. (1), each token in the entire input space $F$ is matched with any tokens, including itself. Our depth completion network benefits from the self-attention mechanism in two folds: 1) it extends the receptive field of our network to the full image in each Transformer layer; 2) as we have embedded each token with both depth and RGB image information, the self-attention mechanism explicitly compare the similarity of each pixel not only by appearance, but also by depth with dot-product operation. Thus, reliable depth information can be broadcasted to the whole image, enabling to correct erroneous pixels.

We boost the representation power of the convolutional path with channel and spatial attention [40]. On the one hand, it helps to model locally accurate attention and reduce noise. On the other hand, due to the semantic gap between convolution and Transformer, the increased modeling capacity by using the attention mechanism enables this path to focus on important features provided by the Transformer layer while suppressing the unnecessary ones. Finally, by concatenating the reshaped feature from the Transformer-based path, we fuse the two paths with a 3 × 3 convolution and send it to the next block or stage.

Taking the proposed JCAT block as the basic unit, we build stages 2-5 with repeated configurations. As reported in Tab. 1, we scale-up the 4 stages in CompletionFormer from tiny, small to base scale. Our results demonstrate the superiority of JCAT design compared to recent Vision Transformers [12, 19, 39] in depth completion task.

### 3.3. Decoder

In the decoder, outputs from each encoding layer are concatenated and further processed by the corresponding decoding layers via skip connections. To accommodate diverse scale features better, the features from previous decoder layer is upsampled to current scale with a deconvolution layer and convolutional attention mechanism [40] is also exploited to strengthen feature fusion in channel and spatial dimensions. Finally, the fused result from the decoder is concatenated with features from stage one and fed to the first convolution layer of the prediction head. Its output is concatenated with the raw feature from RGB and depth embedding module (Sec. 3.1) and sent to another convolution, which is in charge of initial depth prediction $D^0$.

### 3.4. SPN Refinement and Loss Function

Considering that the accurate depth values from the sparse input may not be well preserved after going though U-Net [3, 8], spatial propagation network [18] has been a standard operation for final refinement. Recent work [2, 3, 8, 26] mainly focuses on improving the spatial propagation network from fixed-local to nonlocal propagation. While in our experiments (Tab. 2), we observe that, with our enhanced U-Net backbone, the network is able to provide good depth affinity and thus obtain almost the same accuracy with fixed-local [2, 3] or non-local [26] neighbours for spatial propagation. With regard to CSPN++ [2] consumes more computation cost, we adopt the non-local spatial propagation network [26] (NLSPN) for further refinement. Specifically, let $D^t = (d^t_{u,v}) \in \mathbb{R}^{H \times W}$ denotes the 2D depth map updated by spatial propagation at step $t$, where $d^t_{u,v}$ denotes the depth value at pixel $(u, v)$, and $H, W$ denotes the height and width of the $D^t$ respectively. The propagation of $d^t_{u,v}$ at step $t$ with its non-local neighbors $N^{NL}_{u,v}$ is defined as follows:

$$d^t_{u,v} = w_{u,v}(0,0)d^{t-1}_{u,v} + \sum_{(i,j) \in N^{NL}_{u,v}, i \neq 0, j \neq 0} w_{u,v}(i,j)d^{t-1}_{i,j}, \qquad (2)$$

where $w_{u,v}(i, j) \in (-1, 1)$ describes the affinity weight between the reference pixel at $(u, v)$ and its neighbor pixel at $(i, j)$ and $w_{u,v}(0, 0) = 1 - \sum_{(i,j) \in N_{u,v}, i \neq 0, j \neq 0} w_{u,v}(i, j)$ stands for how much the original depth $d^{t-1}_{u,v}$ will be preserved. Moreover, the affinity matrix $w$ is also outputted by the decoder and modulated by a predicted confidence map from decoder to prevent less confident pixels from propagating into neighbors regardless of how large the affinity is. After $K$ steps spatial propagation, we get the final refined depth map $D^K$.

Finally, following [26], a combined $L_1$ and $L_2$ loss is employed to supervise the network training as follows:

$$L(\hat{D}, D^{gt}) = \frac{1}{|V|} \sum_{v \in V} \left( |\hat{D}_v - D^{gt}_v| + |\hat{D}_v - D^{gt}_v|^2 \right), \qquad (3)$$

where $\hat{D} = D^K$, and $V$ is the set of pixels with valid depth in ground truth $D^{gt}$, and $|V|$ denotes the size of set $V$.

# 4. Experiments

## 4.1. Datasets

**NYUv2 Dataset [33]:** it consists of RGB and depth images captured by Microsoft Kinect [23] in 464 indoor scenes. Following the similar setting of previous depth completion methods [26, 51], our method is trained on 50,000 images uniformly sampled from the training set and tested on the 654 images from the official labeled test set for evaluation. For both training and test sets, the original frames of size $640 \times 480$ are half down-sampled with bilinear interpolation and then center-cropped to $304 \times 228$. The sparse input depth is generated by random sampling from the dense ground truth.

**KITTI Depth Completion (DC) Dataset [36]:** it contains 86 898 training data, 1 000 selected for validation, and 1 000 for testing without ground truth. The original depth map obtained by the Velodyne HDL-64e is sparse, covering about 5.9% pixels. The dense ground truth is generated by collecting LiDAR scans from 11 consecutive temporal frames into one, producing near 30% annotated pixels. These registered points are verified with the stereo image pairs to eliminate noisy values. Since there is no LiDAR return at the top of the image, following [26], input images are bottom center-cropped to $240 \times 1216$ for training, validation and testing phases.

## 4.2. Implementation Details

We implement our model in PyTorch [27] on 4 NVIDIA 3 090 GPUs, using AdamW [20] as optimizer with an initial learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay of 0.01. The batch size per GPU is set to 3 and 12 on KITTI DC and NYUv2 datasets, respectively. On the NYUv2 dataset, we train the model for 72 epochs and decay the learning rate by a factor of 0.5 at epochs 36, 48, 60, 72. For the KITTI DC dataset, the model is trained with 100 epochs, and we reduce the learning rate by half at epochs 50, 60, 70, 80, 90. The supplementary material outlines more details about network parameters.

## 4.3. Evaluation Metrics

Following the KITTI benchmark and existing depth completion methods [26, 51], given the prediction $\hat{D}$ and ground truth $D^{gt}$, we use the standard metrics for evaluation: (1) root mean square error (RMSE); (2) mean absolute error (MAE); (3) root mean squared error of the inverse depth (iRMSE); (4) mean absolute error of the inverse depth (iMAE); (5) mean absolute relative error (REL).

## 4.4. Ablation Studies and Analysis

We assess the impact of the main components of our CompletionFormer on NYUv2 dataset [33]. Following previous methods [16,26], we randomly sample 500 depth pixels from the ground truth depth map and input them along with the corresponding RGB image for network training. Results are reported in Tab. 2.

| | CompletionFormer | RMSE↓<br>(mm) | MAE↓<br>(mm) | Params.↓<br>(M) | FLOPs ↓<br>(G) |
|---|---|---|---|---|---|
| (A) | w/ cascaded connection | 91.5 | 35.7 | 82.6 | 429.6 |
| (B) | w/ parallel connection | **90.0** | **35.0** | 82.6 | 429.6 |
| (C) | w/o Spatial and Channel Attention | 91.1 | 35.5 | **82.5** | **429.4** |
| (D) | w/ dual-branch encoders | 94.0 | 36.4 | 161.0 | 661.4 |

| | Backbone<br>Type | Attention<br>Decoder | Refinement<br>Iterations | RMSE↓<br>(mm) | MAE↓<br>(mm) | Params.↓<br>(M) | FLOPs ↓<br>(G) |
|---|---|---|---|---|---|---|---|
| (E) | ResNet34 [7] | ✗ | 18 | 92.3 | 36.1 | **26.4** | 542.2 |
| (F) | ResNet34 [7] | ✓ | 18 | 91.4 | 35.5 | 28.1 | 582.1 |
| (G) | Swin-Tiny [19] | ✓ | 18 | 92.6 | 36.4 | 38.1 | 634.8 |
| (H) | PVT-Large [39] | ✓ | 18 | 91.4 | 35.6 | 68.3 | 419.8 |
| (I) | MPViT-Base [12] | ✓ | 18 | 91.0 | 35.5 | 83.1 | 1259.3 |
| (J) | CMT-Base [6] | ✓ | 18 | 92.0 | 35.9 | 47.6 | **358.7** |
| (K) | Ours-Small | ✓ | 18 | 90.1 | 35.2 | 82.6 | 439.1 |
| (L) | Ours-Small | ✓ | 6 | **90.0** | 35.0 | 82.6 | 429.6 |
| (M) | Ours-Small | ✓ | CSPN++ | 90.3 | **34.9** | 82.7 | 446.4 |
| (N) | Ours-Tiny | ✓ | 6 | 90.9 | 35.3 | 45.8 | 389.4 |
| (O) | Ours-Base | ✓ | 6 | 90.1 | 35.1 | 146.7 | 499.6 |

Table 2. **Ablation study on NYU Depth v2 [33].** We ablate the settings of our network in the following aspects: the backbone type, the convolutional attention mechanism in decoder and the iterations of NLSPN Refinement module. FLOPs are measured with input resolution $480 \times 640$.

| | SPN | ResNet34 | Swin-Tiny | PVT-Large | MPViT-Base | CMT-Base | Ours-Small |
|---|---|---|---|---|---|---|---|
| RMSE(mm)↓ | ✗ | 106.5 | 106.5 | 106.7 | 100.2 | 108.4 | **99.2** |
| | ✓ | 91.4 | 92.6 | 91.4 | 91.0 | 92.0 | **90.0** |

Table 3. **Ablation study without SPN module.** We report the accuracy of different backbone with/without SPN on NYUv2.

**Cascaded vs Parallel Connection.** We can notice that the cascaded design (A) shows inferior performance compared to parallel style (B). This conclusion is also confirmed in (I) and (J), as CMT-Base gets even worse RMSE (92.0) than MPViT-Base [12] (91.0). It indicates the parallel connection is more suitable for information interaction between streams with different contents and semantics on depth completion task. Thus, we adopt parallel connection as our final scheme.

**Spatial and Channel Attention in JCAT block.** Previous methods combine the Transformer with plain convolutions [6, 12, 25, 28]. Here, we also ablate the case when we disable the spatial and channel attention at the convolutional path of our proposed JCAT block (C). The drop in accuracy (RMSE increases from 90.0 to 91.1) confirms that increasing the capacity of convolutions is vital when combining convolution with Vision Transformer, and it only leads to negligible FLOPs increase (0.2G FLOPs).

**Single- or Dual-branch Encoder.** Similar to previous methods [31, 35], we test the dual-branch architecture, which encodes the RGB and depth information separately (D). For feature communication between two branches, we include the spatial and channel attention mechanism [40] at the end of each stage in the encoder. However, the worse results compared to our single-branch design (B) demonstrates that embedding the multimodal information at the early stage is much more effective and efficient.

| Scanning Lines | Method | RMSE↓ (mm) | MAE↓ (mm) | iRMSE↓ (1/km) | iMAE↓ (1/km) |
|---|---|---|---|---|---|
| 1 | NLSPN | 3507.7 | 1849.1 | 13.8 | 8.9 |
|  | DySPN | 3625.5 | 1924.7 | 13.8 | 8.9 |
|  | Ours-ViT | 3507.2 | 1807.7 | 12.1 | 7.8 |
|  | Ours | **3250.2** | **1582.6** | **10.4** | **6.6** |
| 4 | NLSPN | 2293.1 | 831.3 | 7.0 | 3.4 |
|  | DySPN | 2285.8 | 834.3 | 6.3 | 3.2 |
|  | Ours-ViT | 2241.2 | 795.9 | 5.8 | 2.9 |
|  | Ours | **2150.0** | **740.1** | **5.4** | **2.6** |
| 16 | NLSPN | 1288.9 | 377.2 | 3.4 | 1.4 |
|  | DySPN | 1274.8 | 366.4 | 3.2 | 1.3 |
|  | Ours-ViT | 1268.9 | 360.7 | 3.3 | 1.3 |
|  | Ours | **1218.6** | **337.4** | **3.0** | **1.2** |
| 64 | NLSPN | 889.4 | 238.8 | 2.6 | 1.0 |
|  | DySPN | 878.5 | 228.6 | **2.5** | 1.0 |
|  | Ours-ViT | 872.0 | 226.2 | **2.5** | 1.0 |
|  | Ours | **848.7** | **215.9** | **2.5** | **0.9** |

Table 4. **Ablation study on scanning lines of LiDAR sensor on KITTI DC [36] datasets.** Ours-ViT denotes that only the Transformer layer is enabled in our proposed block.

**Comparisons with General Feature Backbones.** In RMSE, our novel CompletionFormer in small scale (K) outperforms pure CNNs based method (F) and those pure Transformer-based variants (G, H) counting comparable FLOPs with respect to our model. In particular, compared to the recent MPViT-Base [12] (I) and CMT-Base [6] (J) which also integrate CNNs and Transformer for feature extraction, our network achieves higher accuracy and much lower computational overhead (429.6G FLOPs) than MPViT-Base (1259.3G FLOPS).

**Decoder.** Compared to our baseline (E), *i.e.*, NLSPN [26], introducing spatial and channel attention for multiscale feature fusion in the decoder (F) further improves the results, RMSE drops from 92.3 to 91.4.

**SPN Refinement Iterations.** Our network (L) requires as few as 6 iterations for SPN refinement to converge to the best result. Compared to baseline (E) which requires 18 iterations, our enhanced U-Net has collected information from the whole image and thus doesn't need large iterations to propagate to long distance. Even when the non-local refinement in NLSPN replaced with fixed-local neighbors in CSPN++ [2] (M), the accuracy remains almost the same. It indicates that our CompletionFormer can learn good affinity locally and globally, thus helping to soften the problem raised by the limited and fixed range of aggregation in CSPN++ [2].

**Model Scales.** Our models in various scales (L, N, O), benefiting from local and global cues, achieve significant improvement compared to the pure CNN-based baseline (E), while counting fewer FLOPs. To trade-off between accuracy and efficiency, we select our model in small scale (L) as our final architecture for the remaining experiments.

**With/without SPN refinement.** To conclude, in Tab. 3 we show the results achieved by different backbones with and without SPN refinement. Ours yields the most accurate results in both cases.

| Method |  | RMSE↓ (m) |  |  |  |  |
|---|---|---|---|---|---|---|
|  |  | PackNet-SAN | GuideNet | NLSPN | Ours-ViT | Ours |
| Sample Number | 0 | - | - | 0.562 | 0.544 | **0.490** |
|  | 50 | - | - | 0.223 | 0.218 | **0.208** |
|  | 200 | 0.155 | 0.142 | 0.129 | 0.130 | **0.127** |
|  | 500 | 0.120 | 0.101 | 0.092 | 0.091 | **0.090** |

Table 5. **Sparsity Studies on the NYUv2 Dataset.** Evaluation with 0, 50, 200 and 500 samples.

| Method | KITTI DC |  |  |  | NYUv2 |  |
|---|---|---|---|---|---|---|
|  | MAE↓ (mm) | iMAE↓ (1/km) | iRMSE↓ (1/km) | RMSE↓ (mm) | RMSE↓ (m) | REL↓ |
| CSPN [3] | 279.46 | 1.15 | 2.93 | 1019.64 | 0.117 | 0.016 |
| DeepLiDAR [29] | 226.50 | 1.15 | 2.56 | 758.38 | 0.115 | 0.022 |
| GuideNet [35] | 218.83 | 0.99 | 2.25 | 736.24 | 0.101 | 0.015 |
| NLSPN [26] | 199.59 | 0.84 | 1.99 | 741.68 | 0.092 | **0.012** |
| PENet [8] | 210.55 | 0.94 | 2.17 | 730.08 | - | - |
| ACMNet [49] | 206.09 | 0.90 | 2.08 | 744.91 | 0.105 | 0.015 |
| TWISE [9] | 195.58 | 0.82 | 2.08 | 840.20 | 0.097 | 0.013 |
| RigNet [45] | 203.25 | 0.90 | 2.08 | 712.66 | **0.090** | 0.013 |
| GuideFormer [31] | 207.76 | 0.97 | 2.14 | 721.48 | - | - |
| DySPN [16] | 192.71 | 0.82 | **1.88** | 709.12 | **0.090** | **0.012** |
| Ours ($L_1$) | **183.88** | **0.80** | 1.89 | 764.87 | - | - |
| Ours ($L_1$+$L_2$) | 203.45 | 0.88 | 2.01 | **708.87** | **0.090** | **0.012** |

Table 6. **Quantitative evaluation on KITTI DC and NYUv2.**

## 4.5. Sparsity Level Analysis

Depth maps captured by sensors such as Microsoft Kinect [23] and Velodyne LiDAR sensor are unevenly distributed and often contain outliers. To compare the effectiveness of our model with current state-of-the-art methods [5, 16, 26, 35] when dealing with this challenging input depth, we manually generate sparse data at different settings for network training and testing. As for noise, since captured depth maps are often corrupted by sensor noise or displacement between RGB camera and depth sensor [29], we do not add extra noise to the input depth for experiments.

**Outdoor Scene.** We conduct exhaustive experiments on the KITTI DC dataset [36] to illustrate the robustness of our CompletionFormer when input with different sparsity levels. For all experiments, we train on randomly selected 10 000 RGB and LiDAR pairs from official training data (due to resource constraints) and test on the selected validation dataset provided by KITTI. Furthermore, following [9], we sub-sample the raw LiDAR points captured by Velodyne HDL-64e in azimuth-elevation space into 1, 4, 16 and 64 lines to simulate the LiDAR-like patterns. All methods have been fully **retrained** using the official code with variable lines of LiDAR inputs (For DySPN [16], as no public code available, the results have been provided to us by the author using the same selected training list). To get a throughout understanding of the local details encoded by convolution and global context gathered by Transformer, we present the results estimated by pure CNN-based methods (*i.e.* NLSPN and DySPN), a fully Transformer-based variant of our network (Ours-ViT) and our CompletionFormer complete architecture (Ours), which integrates both paradigms. As reported in Tab. 4, thanks to the global correlations built by Transformers, our model relying on these blocks only

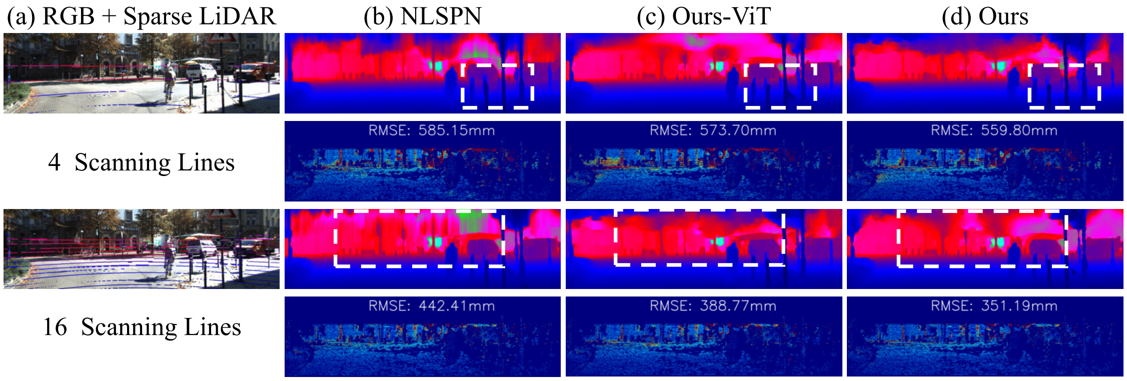


Figure 4. **Qualitative results on KITTI DC selected validation dataset with 4 and 16 LiDAR scanning lines.** We attach the subsampled LiDAR lines to the corresponding RGB image for better visualization. Ours-ViT denotes that only the Transformer layer is enabled in our proposed block. A colder color in depth and error maps denotes a lower value.

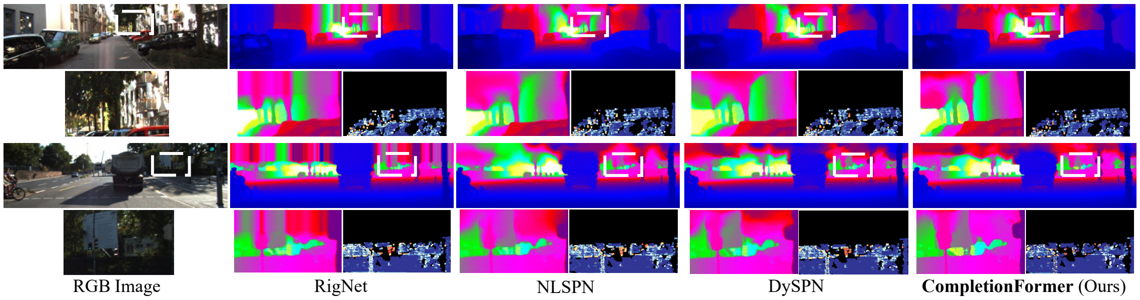


Figure 5. **Qualitative results on the KITTI depth completion test set.** Comparisons of our method against state-of-the-art methods including RigNet [45], NLSPN [26], DySPN [16] are presented. We provide RGB images, dense predictions, zoom-in views of challenging areas and corresponding error maps for better visualization.

(Ours-ViT) exhibits better results in all metrics as the LiDAR points get sparser. However, solely using Transformer layers makes it difficult to distinguish the objects from the background, as shown in Fig. 4. By coupling the local features and global representations, our complete model (Ours) significantly decreases the errors in all metrics.

**Indoor Scene.** On the NYUv2 dataset [33], we randomly sample 0, 50, 200 and 500 points from the ground truth depth map to mimic different depth sparsity levels while keeping the ground truth depth used for supervision unchanged. Both our model and NLSPN with publicly available code are retrained for a fair comparison, while the results of GuideNet [35] and PackNet-SAN [5] are taken from the original papers. In Tab. 5, CompletionFormer with both CNNs and Transformers consistently outperforms all other methods in any cases. Qualitative results are provided in the supplementary material.

### 4.6. Comparison with SOTA Methods

This section comprehensively assesses the performance of state-of-the-art (SOTA) methods. On **indoor**, *i.e.* NYUv2 dataset [33], CompletionFormer achieves the best results as reported in Tab. 6. When moving to **outdoor** dataset, MAE, iMAE, RMSE and iRMSE are adopted for benchmark on KITTI depth completion (DC) dataset [36]. Empirically, our model trained with only $L_1$ loss achieves the best results on two among four metrics (MAE and iMAE). By jointly minimizing $L_1$ and $L_2$ losses, CompletionFormer ranks first on RMSE metric among published methods. Qualitative results on the KITTI DC test dataset are provided in Fig. 5. By integrating convolutions and Transformers, our model performs better near depth missing areas (*e.g.* the zoom-in visualization on the second and fourth rows), textureless objects (*e.g.* the cars on the first row) and small objects (*e.g.* the pillars and tree stem far in the distance, on second and fourth rows).

## 5. Conclusion and Limitations

This paper proposed a single-branch depth completion network, CompletionFormer, seamlessly integrating convolutional attention and Transformers into one block. Extensive ablation studies demonstrate the effectiveness and efficiency of our model in depth completion when the input is sparse. This novel design yields state-of-the-art results on indoor and outdoor datasets. Currently, CompletionFormer runs at about 10 FPS: decreasing its runtime further to meet real-time requirements will be our future work.

## References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *AAAI*, volume 34, pages 10615–10622, 2020.

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

[4] 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. *ArXiv preprint arXiv:2010.11929*, 2020.

[5] Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *CVPR*, pages 11078–11088, 2021.

[6] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *CVPR*, pages 12175–12185, June 2022.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016.

[8] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *ICRA*, pages 13656–13662. IEEE, 2021.

[9] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *CVPR*, pages 2583–2592, 2021.

[10] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, pages 9772–9781, 2021.

[11] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *CVPR Workshops*, pages 1–10, 2017.

[12] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *CVPR*, 2022.

[13] Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. *ArXiv preprint arXiv:2203.11483*, 2022.

[14] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *ArXiv preprint arXiv:2203.14211*, 2022.

[15] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X Creighton, Russell H Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *ICCV*, pages 6197–6206, 2021.

[16] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *AAAI*, 2022.

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

[18] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. *NeurIPS*, 2017.

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

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

[23] Microsoft. Kinect for windows. https://developer.microsoft.com/en-us/windows/kinect/.

[24] Danish Nazir, Marcus Liwicki, Didier Stricker, and Muhammad Zeshan Afzal. Semattnet: Towards attention-based semantic aware guided depth completion. *arXiv preprint arXiv:2204.13635*, 2022.

[25] Xuran Pan, Chunjiang Ge, Rui Lu, Shiji Song, Guanfu Chen, Zeyi Huang, and Gao Huang. On the integration of self-attention and convolution. In *CVPR*, pages 815–825, 2022.

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

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*, pages 8024–8035. Curran Associates, Inc., 2019.

[28] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *ICCV*, pages 367–376, October 2021.

[29] 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 *CVPR*, pages 3313–3322, 2019.

[30] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021.

[31] Kyeongha Rho, Jinsung Ha, and Youngjung Kim. Guideformer: Transformers for image guided depth completion. In *CVPR*, pages 6250–6259, 2022.

[32] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

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

[34] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Siow Mong Goh, and Hongyuan Zhu. Craft: Cross-attentional flow transformers for robust optical flow. In *CVPR*, 2022.

[35] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE TIP*, pages 1116–1129, 2020.

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

[37] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *MVA*, pages 1–6. IEEE, 2019.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[39] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021.

[40] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, pages 3–19, 2018.

[41] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 34, 2021.

[42] Xin Xiong, Haipeng Xiong, Ke Xian, Chen Zhao, Zhiguo Cao, and Xin Li. Sparse-to-dense depth completion revisited: Sampling strategy and graph construction. In *ECCV*, pages 682–699. Springer, 2020.

[43] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *ICCV*, pages 9981–9990, 2021.

[44] Zheyuan Xu, Hongche Yin, and Jian Yao. Deformable spatial propagation networks for depth completion. In *ICIP*, pages 913–917. IEEE, 2020.

[45] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Baobei Xu, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. *ECCV*, 2022.

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

[47] Yongchi Zhang, Ping Wei, Huan Li, and Nanning Zheng. Multiscale adaptation fusion networks for depth completion. In *IJCNN*, pages 1–7, 2020.

[48] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. In *3DV*, 2022.

[49] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *TIP*, pages 5264–5276, 2021.

[50] Yiqi Zhong, Cho-Ying Wu, Suya You, and Ulrich Neumann. Deep rgb-d canonical correlation analysis for sparse depth completion. *NeurIPS*, 2019.

[51] Yufan Zhu, Weisheng Dong, Leida Li, Jinjian Wu, Xin Li, and Guangming Shi. Robust depth completion with uncertainty-driven loss functions. In *AAAI*, 2022.