Adaptive Context-Aware Multi-Modal Network for Depth Completion

Shanshan Zhao Mingming Gong Huan Fu Dacheng Tao

Abstract

Depth completion aims to recover a dense depth map from the sparse depth data and the corresponding single RGB image. The observed pixels provide the significant guidance for the recovery of the unobserved pixels' depth. However, due to the sparsity of the depth data, the standard convolution operation, exploited by most of existing methods, is not effective to model the observed contexts with depth values. To address this issue, we propose to adopt the graph propagation to capture the observed spatial contexts. Specifically, we first construct multiple graphs at different scales from observed pixels. Since the graph structure varies from sample to sample, we then apply the attention mechanism on the propagation, which encourages the network to model the contextual information adaptively. Furthermore, considering the mutli-modality of input data, we exploit the graph propagation on the two modalities respectively to extract multi-modal representations. Finally, we introduce the symmetric gated fusion strategy to exploit the extracted multi-modal features effectively. The proposed strategy preserves the original information for one modality and also absorbs complementary information from the other through learning the adaptive gating weights. Our model, named Adaptive Context-Aware Multi-Modal Network (ACMNet), achieves the state-of-the-art performance on two benchmarks, i.e., KITTI and NYU-v2, and at the same time has fewer parameters than latest models. Our code is available at: https://github.com/sshan-zhao/ACMNet.

1. Introduction

Depth information is crucial for 3D vision tasks, *e.g.*, 6D object pose estimation [1], 3D object detection [2], and human pose estimation [3]. To complete these tasks, various depth sensors such as LiDAR have been invented to acquire depth information. However, current depth sensors are not able to obtain dense maps for outdoor scenes, which are essential in various applications, especially autonomous driving. Therefore, depth completion from sparse depth maps[1] and RGB images has attracted intensive attention. Depth completion is a challenging problem because the depth values obtained by sensors are highly sparse and irregularly spaced. For example, in the KITTI dataset [4], there are only 5.9% pixels with depth information obtained by the Velodyne HDL-64e (64 layers) LiDAR in the whole image space, as shown in Figure 1. Traditional methods [5, 6, 7] rely on handcrafted features and global constraints on the output depth values, which are inaccurate. Recent studies [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18] have demonstrated great advantages of deep Convolutional Neural Networks (CNNs) on depth completion. By extending the convolutional operation with sparsity-invariance [9, 19, 20] or introducing more geometric information [21, 22], these deep methods can achieve way better performance than traditional methods.

In spite of the encouraging progress, existing depth completion methods suffer from a significant issue, which limits the depth completion performance. Specifically, the conventional convolutional operation applies kernels with regular structure (*e.g.,* 3×3) at all locations, which ignores the fact that the observed depth values are irregularly distributed in a sparse depth map and associates limited observed contexts for the unobserved, as shown in Figure 2. Thus, CNN-based methods are not adaptive to the pattern of observed spatial contextual information in a sparse depth map, resulting in a sub-optimal prediction of depth in unobserved locations.

To address this issue and further boost depth completion accuracy, we propose an Adaptive Context-Aware Multi-Modal Network (ACMNet, shown in Figure 3). Firstly, inspired by recent works on point cloud analysis [23], we model the observed contextual information adaptively by applying attention based graph propagation within multiple graphs constructed from observed pixels. Based on the efficient graph propagation, the model can associate the spatial context with observed depth values and then enhance the features of the unobserved pixels. To illustrate this, we provide a simple example in Figure 2. Compared to the sole convolutional operations, the proposed graph propagation (followed by a convolution) can make the unobserved pixels capture more related observed contextual information.

Furthermore, since we have multi-modality data, we need to reconsider the novel graph propagation in a multi-

[1]The sparse depth map is generated by projecting the LiDAR data to the image plane, and the value in locations without depth information is 0.

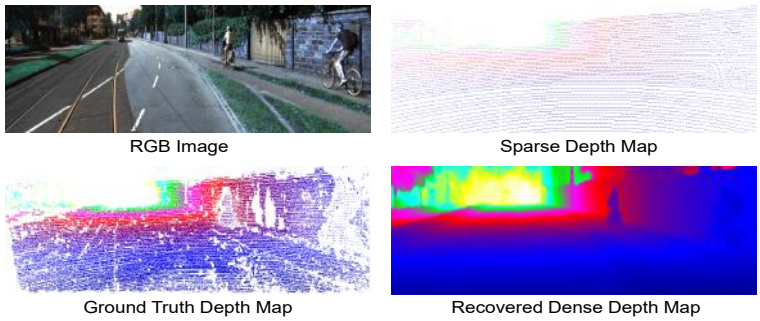


Figure 1: Depth Completion from LiDAR Data and RGB Image by ACMNet. Top: RGB image and sparse LiDAR data; Bottom: ground truth depth map and dense depth map obtained by our approach.

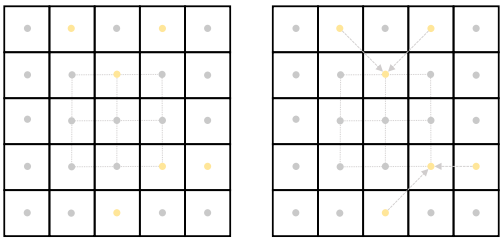


Figure 2: Illustration of convolution and graph propagation. Left: convolution (3×3 kernel); Right: graph propagation (2-nearest neighbours) and convolution (3×3 kernel). The observed pixels are marked by the yellow, while the unobserved are marked by the gray.

modal setting. Firstly, to better learn the relationship between observed pixels (nodes), we use the co-attention mechanism [24] to propagate the multi-modal information of observed pixels in a symmetric structure. This step is conducted in the encoder to extract multi-scale and multi-modal features. However, this mechanism does not consider the fusion of multi-modal contextual information. A simple way to fuse the multi-modal data is by applying the simple concatenation or element-wise summation operation on the extracted feature maps, which was used by most of the existing works, *e.g.,* [11, 21]. However, this type of fusion strategy cannot fully explore the heterogeneity of the two modalities. To address the issue, we further present the symmetric gated fusion strategy to combine the depth and RGB information in the decoder. In specific, the presented fusion strategy consists of two branches. One branch focuses on fusing the RGB information as supplementary into the depth information through learning an adaptive gating function, and the other one does the opposite. Therefore, each branch can maintain its own information and benefit from supplementary information from the other. Benefiting from the adaptive co-attention guided graph propagation and symmetric gated multi-modal feature fusion, our ACMNet is able to generate high-quality dense depth maps. In summary, our main contributions are:

- We introduce the co-attention guided graph propagation to our depth completion network, which is adaptive to the sparsity patterns of sparse depth input and thus enables the unobserved pixels to capture useful observed contextual information more effectively.
- To fuse the multi-modal contextual information efficiently, we further present the symmetric gated fusion strategy, which can learn the heterogeneity of the two modalities adaptively.
- We demonstrate the effectiveness of ACMNet on two benchmarks, *i.e.,* KITTI Depth Completion Dataset [4] and NYU-v2 Dataset [25].

2. Related Work

Depth Completion. Traditional approaches solve the depth completion problem by formulating the task as an energy function optimization problem [5, 26, 6, 7]. However, these works showed some limitations in performance due to the employment of hand-crafted features.

Currently, CNNs have been a dominant solution for depth completion [21, 27, 28, 29, 30, 31, 32, 19, 33, 34, 17, 18, 35, 36], outperforming traditional methods by a wide margin. In specific, to learn representations of the irregular and sparse LiDAR data, Uhrig *et al*. [9] proposed the sparsity-invariant convolutional operation. Following this work, some variants of the sparse convolution are introduced [20, 19, 17]. In the case of additional RGB data, Jaritz *et al*. [11] showed that the late fusion strategy outperformed the early fusion. Ma *et al*. [31] utilized self-supervised learning on sparse LiDAR data coupled with the stereo image pair to mitigate the need for ground truth dense

depth. Yang *et al.* [37] exploited the Conditional Prior Network [32] to learn a depth prior on synthetic images. Additionally, there are also a bunch of works [8, 30, 22] exploring other cues. For example, Zhang *et al*. [8] trained a network to predict local surface normals for indoor scene depth completion, and later an extension for outdoor scenes was introduced in their latest work [21]. Similarly, Xu *et al*. [22] also explored the surface normal information to improve the performance by introducing a diffusion module. Cheng *et al.* [28, 34] proposed to learn affinities between adjacent pixels for the spatial propagation of the depth information. Following the two works, a recent work [38] improved the propagation strategy through concentrating on the non-local neighbors and introducing a learnable affinity normalization. Inspired by the guided image filtering, Tang *et al.* [29] designed a guided convolution module, which generates dynamic spatially-variant kernels using the image features, to extract the depth image features. In comparison, a recent work [39] proposed to dynamically learn the filter by applying the Graph Neural Network (GNN) [40] on the graph constructed from the predicted dense depth map. In contrast to these approaches, which paid little attention to the modelling of the multi-modal contexts, our work mainly aims at making unobserved pixels capture more useful observed contextual information from the input multi-modal data. Additionally, it is worth pointing out that although the latest work [39] also exploits the graph models, there are many differences between it and ours. For example, it aims to consider the neighborhood relationship of the points in the 3D space through constructing a 3D graph from the dense depth map, which is obtained using a deep model. To arrive at this, it applies the dynamic kernel, which is learned through using a typical GNN model on the constructed graph, on the dense features at $1/8$ of original scale. In comparison, in this paper we study the propagation of the contexts with observed depth values at multiple scales in a multi-modal setting to enhance the features of the unobserved pixels.

Monocular Depth Estimation. From approaches based on probabilistic graphical models (*e.g.*, MRFs) with handcrafted features [41, 42] to the deep learning-based [43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54], the improvement of performance for monocular depth estimation has been pushed forward. Eigen *et al.* [46] were the first to develop deep models for depth estimation. Following their work, a lot of supervised approaches [55, 56, 43, 57] have been proposed. However, these methods rely on large quantities of ground truth depth data, which is hard to acquire. To address this issue, Garg *et al.* [48] and Godard *et al.* [45] proposed to predict depth maps from stereo pair images by exploring unsupervised cues, while some recent works tried to utilize synthetic data [58, 59, 60, 61, 62] based on the domain adaptation technique [63].

Graph-based Models. Conventional deep learning modules, such as CNNs, do not perform well on graphs. To model the graph data efficiently, Graph Models have been applied on various computer vision tasks, such as action recognition [64, 65], point cloud analysis [66, 23], few-shot image classification [67, 68], and person re-identification [69]. Graph Models are able to learn the representation of each target node by propagating its neighborhood information in a data-driven way and thus associate the contextual information. In this work, we design an attention-based graph propagation module and then extend it to the co-attention guided graph propagation for multi-modal data, which is capable of learning an efficient multi-modal representation for the input data through encouraging the adaptive contextual interactions.

Multi-modal Information Fusion. Multi-modal information fusion has been studied in various computer vision tasks, such as visual question answering [70], video action recognition [71], 3D object detection [72], and many more. A simple approach to fuse the multi-modal data is applying concatenation or summation operation into the input data or extracted feature maps [11, 71]. However, for a specific task, different modalities often provide different information, and therefore, the naive fusion strategy might fail to combine them effectively. To address this issue, some works, *e.g.,* [72, 73], proposed to exploit the attention mechanism to improve the performance. As for depth completion, current works mainly employed the naive fusion strategy. In fact, both naive strategy and attention based approaches fuse the multi-modal features in a single way, which is not enough to extract complementary information and then limits the performance. In contrast, we present the symmetric gated fusion strategy consisting of two fusion paths, each of which only focuses on one modal and extracts useful information adaptively from the other.

3. Our Approach

3.1. Problem Formulation

Our goal is to recover a dense depth map from the observed sparse depth data and a single RGB image. Mathematically, given a set of paired samples $\{(\mathbf{X}_S, \mathbf{X}_I)_i\}_{i=0}^{N-1}$, we expect to learn a mapping function $f(\cdot)$ that satisfies $\mathbf{Y} = f(\mathbf{X}_S, \mathbf{X}_I)$, where $\mathbf{X}_S \in \mathbb{R}^{H\times W}$, $\mathbf{X}_I \in \mathbb{R}^{3\times H\times W}$, and $\mathbf{Y} \in \mathbb{R}^{H\times W}$ represent the sparse depth map, the RGB image, and the ground truth depth map, respectively. To achieve this target, we develop a high-performing depth completion network (ACMNet) building on two novel modules, including a co-attention guided graph propagation module (CGPM) and a symmetric gated fusion module (SGFM), as shown in Figure 3. In specific, we first employ a series of CGPMs to effectively extract contextual information from $\mathbf{X}_S$ and $\mathbf{X}_I$. Then we exploit SGFMs to learn the

complementarity between contextual representations from multi-modalities. In the following, we will present our network architecture and the proposed modules in detail.

3.2. Network Architecture

Our overall network architecture follows a two-stream encoder-decoder fashion as previously [30, 31, 11, 13], but with the improvement by integrating the novel CGPM and SGFM. We show the whole framework in Figure 3, and briefly explain the encoder and the decoder right here.

Encoder. The encoder targets learning discriminative multi-scale features from both the sparse depth and the RGB image. While researchers reached a consensus that standard convolutional operations can perform well in the image data, how to extract rich information from observed spatial contexts is still an open problem due to the extreme sparsity [9, 20, 19, 11, 30, 29, 22]. In this paper, we show that the proposed CGPM has the potential to capture the related contextual information from the observed pixels with various patterns in an adaptive manner through learning dynamic weights of the relationship between adjacent nodes in the constructed graph. Specifically, our encoder consists of two conventional convolutional layers followed by a stack of CGPMs. The encoded features at each scale $\{\mathbf{F}_S^l\}_{l=1}^L$ and $\{\mathbf{F}_I^l\}_{l=1}^L$ can be computed as:

$$(\mathbf{F}_S^l, \mathbf{F}_I^l) = f_e^l(\mathbf{F}_S^{l-l}, \mathbf{F}_I^{l-1}), \mathbf{F}_S^l, \mathbf{F}_I^l \in \mathbb{R}^{C^l \times \frac{H}{2^l} \times \frac{W}{2^l}}, \quad (1)$$

where $l = 1, 2, ..., L$, and f_e^l denotes the CGPM at level l, and $\mathbf{F}_S^0$ and $\mathbf{F}_I^0$ are the outputs of the beginning convolutional layers.

Decoder. The decoder aims to predict depth values of unobserved pixels in $\mathbf{X}_S$ given multi-scale and multi-modal features generated by the encoder mentioned above. To this end, one of the commonly studied problems is how to take full advantage of multi-modal representations. A straightforward idea is to directly concatenate or sum features progressively at different scales [11, 21]. However, as analyzed before, these naive fusion strategies fail to model the complementary information between multiple modalities satisfyingly. To alleviate the issue, we propose an adaptive symmetric gated fusion strategy to fuse the multi-modal contextual representations in a parallel structure. In specific, we design two parallel branches in the decoder, *i.e.,* the depth and image branches. The depth branch preserves discriminative information of the sparse depth modality and meanwhile adaptively captures comprehensive information from the image model through learning dynamic gating weights, and vise versa for the image branch. The overall decoder architecture is described as follows.

As shown in Figure 3, at the beginning of the decoder, we feed $\mathbf{F}_S^L$ coupled with $\mathbf{F}_I^L$ into the first SGFM to generate the fused feature $\mathbf{Q}_{SI}^L$ and $\mathbf{Q}_{SI}^{L,\uparrow}$, which is acquired by upsampling $\mathbf{Q}_{SI}^L$ through one deconvolutional layer. At the following levels l from $L-1$ to 0, $\mathbf{Q}_{SI}^{l+1,\uparrow}$, $\mathbf{F}_S^l$ and $\mathbf{F}_I^l$ are fed into the SGFM at level l together. Similarly, we can obtain the intermediate features $\mathbf{Q}_{IS}^l$ in the image branch. The procedure can be expressed as:

$$\begin{aligned}(\mathbf{Q}_{SI}^L, \mathbf{Q}_{IS}^L, \mathbf{Q}_{SI}^{L,\uparrow}, \mathbf{Q}_{IS}^{L,\uparrow}) &= f_d^L(\mathbf{F}_S^L, \mathbf{F}_I^L),\\ (\mathbf{Q}_{SI}^l, \mathbf{Q}_{IS}^l, \mathbf{Q}_{SI}^{l,\uparrow}, \mathbf{Q}_{IS}^{l,\uparrow}) &= f_d^l(\mathbf{Q}_{SI}^{l+1,\uparrow}, \mathbf{Q}_{IS}^{l+1,\uparrow}, \mathbf{F}_S^l, \mathbf{F}_I^l),\end{aligned} \quad (2)$$

where $l = L-1, L-2, ..., 0$, and f_d^l represents the SGFM.

Finally, we present two methods, *i.e.,* end-integration and feature-integration, to combine the two branches to obtain the final recovered dense depth map, which will be described in detail in Sec. 3.5.

3.3. Co-Attention Guided Graph Propagation (CGPM)

The proposed CGPM is composed of a residual connection and a co-attention guided graph propagation module. First, we introduce the basic graph propagation module, which is employed in CGPM. In specific[2], given the spatial position set $P = \{p_0, p_1, ..., p_{n-1}\}$ of n pixels with observed depth values, we define a graph $G(V, E)$, where V is the vertex (or node) set corresponding to P, and $E \subseteq |V| \times |V|$ is the edge set. For a vertex i, we connect it to the k nearest neighbour $\mathcal{N}_i$ according to the spatial locations. Note that, we build an individual graph for the CGPM at each scale. Thus, to obtain a specific P^l at level l, which is in lower resolution, we generate $\mathbf{X}_S^l$ by applying max-pooling based down-sampling operation on $\mathbf{X}_S^{l-1}$. The graph's construction process can be found in Figure 4. In the following, we first introduce the basic attention guided graph propagation component at level l by taking the image stream as an example, then present the full CGPM.

Given the graph G and the input feature maps $\mathbf{F}_I^{l-1}$, we expect to learn discriminative $\mathbf{F}_I^l$ by both adaptively encoding the contextual information of scenes and exploiting guidance for unobserved pixels from observed pixels. Specifically, we exploit two efficient stages, *i.e.,* adaptive feature propagation within observed pixels and feature enhancement of unobserved pixels.

At the first stage, we employ one standard convolutional layer to extract $\mathbf{F}_I'$ from $\mathbf{F}_I^{l-1}$, and denote $\mathbf{F}_{Io}' \in \mathbb{R}^{n \times C}$ as the feature vectors of all the nodes in G. Then, we adaptively aggregate neighboured information for each node i in G as:

$$\begin{aligned}\alpha^{i,j} &= \frac{exp(\mathbf{W}^{i,j})}{\sum_{k \in \mathcal{N}_o^i} exp(\mathbf{W}^{i,k})},\\ \mathbf{F}_{Io}''^i &= \sum_{j \in \mathcal{N}_o^i} \alpha^{i,j} \mathbf{F}_{Io}'^i,\end{aligned} \quad (3)$$

[2]In the following part, we deprecate the scale indexes l to simplify our presentation in some cases.

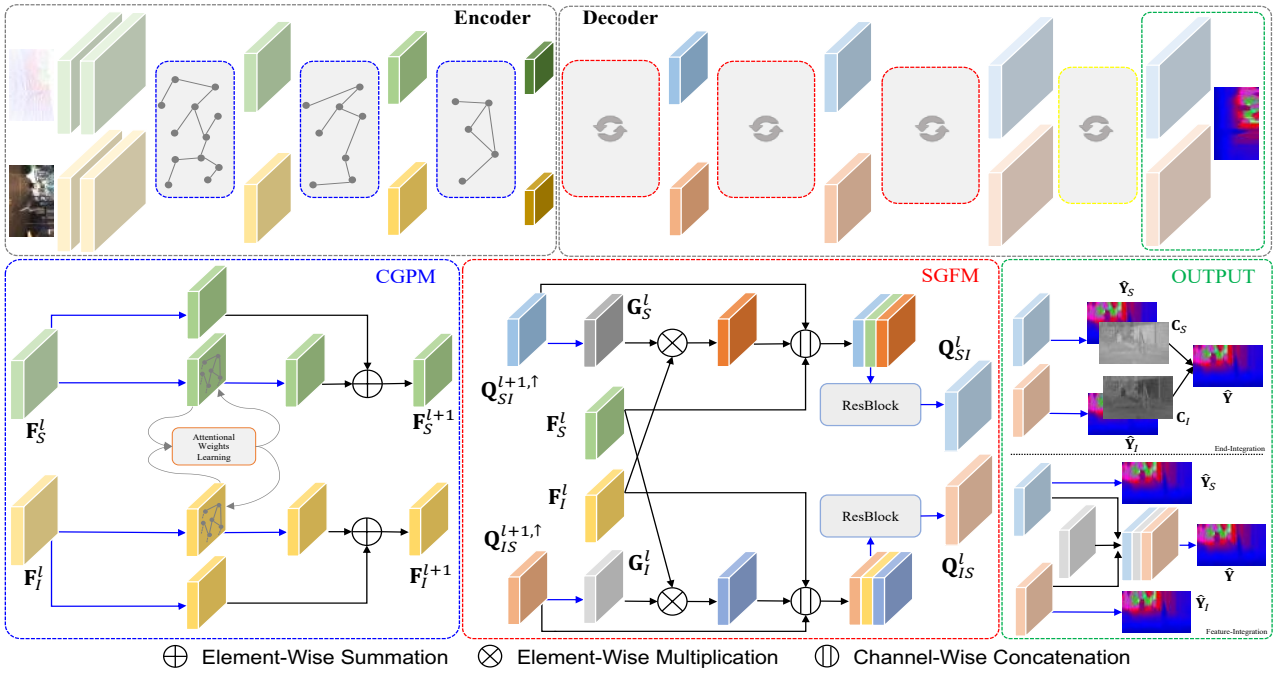


Figure 3: The proposed ACMNet in this paper. Left upper part: Encoder; Right upper part: Decoder. In encoding stage, we extract multi-scale multi-modal features using a stack of CGPMs (Marked by blue dotted box, Sec. 3.3), and the adaptive attentional weights are learnt from spatial locations, depth features and RGB features. In decoding stage, we fuse the multi-modal features progressively by exploiting the SGFMs, represented by red dotted boxes (Sec. 3.4). Lastly, final output is calculated from the dense maps and confidence maps produced by the two branches of the decoder or predicted using the intermediate fused features maps, shown in the green dotted box (Sec. 3.5). Note that, the yellow dotted box denotes that there is no ResBlock behind the initial fusion (see Sec. 3.4) in the SGFM. Blue arrow: convolution; Gray arrow: graph propagation; Black arrow: summation/multiplication/concatenation.

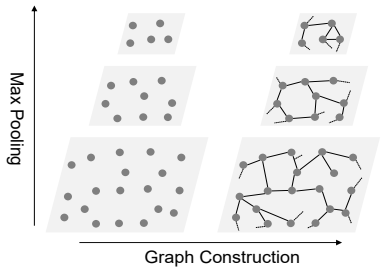


Figure 4: Graph Construction. At each scale, we use k (*e.g.*, k is 3 in this example) nearest neighbour to construct the graph from the observed pixels, represented by gray circles.

where $\alpha^{i,j}$ is the computed attentional weight, and $\mathbf{W}^{i,j}$ is the adaptive weight between neighboured nodes i and j. Here, inspired by the works [74, 66, 75] on point cloud, we exploit the self-attention mechanism [76] to learn $\mathbf{W}^{i,j}$ adaptively by modelling the relationship between the connected nodes. Mathematically, the mapping function f_w between $\mathbf{F}'_{Io}$ and $\mathbf{W}^{i,j}$ can be expressed as:

$$\mathbf{W}^{i,j} = f_w([\Delta p^{i,j} || \Delta \mathbf{F}'^{i,j}_{Io}]), j \in \mathcal{N}^i_o, \tag{4}$$

where $[\cdot||\cdot]$ represents the concatenation operation, $\Delta p^{i,j} = p^j - p^i$ and $\Delta \mathbf{F}'^{i,j}_{Io} = \mathbf{F}'^{j}_{Io} - \mathbf{F}'^{i}_{Io}$ denote the spatial and feature distances between node i and j, respectively. The f_w is implemented by a two-layer MLP, the first one followed by one LeakyReLU activation function [77]. Note that, permutation variant operations like convolution are not allowed here due to the unordered input. After obtaining $\mathbf{F}''_{Io}$, the features of unobserved pixels are enhanced by a standard convolutional operation. In addition, a residual connection [78] is also utilized to preserve early information. We can use the same algorithm to conduct propagation in the depth stream.

As shown in Figure 3, in the CGPM in our encoder, we learn the adaptive weights $\mathbf{W}_S$ and $\mathbf{W}_I$ by considering both information from the image stream and the sparse depth stream, inspired by the co-attention mechanism [24].

Therefore, in each CGPM, Eq. 4 can be re-written as:

$$\begin{aligned}\mathbf{W}_S^{i,j} &= f_{Sw}([\Delta p^{i,j}||[\Delta \mathbf{F}_{So}^{'i,j}||\Delta \mathbf{F}_{Io}^{'i,j}]]), j \in \mathcal{N}_o^i, \\ \mathbf{W}_I^{i,j} &= f_{Iw}([\Delta p^{i,j}||[\Delta \mathbf{F}_{Io}^{'i,j}||\Delta \mathbf{F}_{So}^{'i,j}]]), j \in \mathcal{N}_o^i.\end{aligned} \tag{5}$$

3.4. Symmetric Gated Fusion (SGFM)

For obtained features $\mathbf{F}_S$ and $\mathbf{F}_I$, we develop an effective fusing strategy to adaptively absorb complementary information from the multi-modal contextual representations. For example, depth features encode the scene geometry structure, *e.g.,* the distance from the camera to partial spatial locations. It contributes to inferring the depth of unobserved locations directly. In addition, RGB features contain semantic information and provide prior appearance knowledge of unobserved pixels. Instead of concatenating or summing them together directly with or without attention mechanism, we exploit the proposed SGFM with a symmetric structure, as shown in Figure 3. More specifically, at the beginning of the decoder, we employ the convolutional operation followed by a Sigmoid function on $\mathbf{F}_S^L$ to generate the adaptive gating weight $\mathbf{G}_S^L$. By applying the adaptive attention mechanism, the network can absorb meaningful information from the RGB branch and filter out the unrelated. Then we we feed the initial fused feature $[\mathbf{F}_S^L||\mathbf{G}_S^L * \mathbf{F}_I^L]$ into the Residual Block (*abbr.* ResBlock) [78] to obtain the final fused features $\mathbf{Q}_{SI}^L$, which is then fed into a deconvolutional layer to generate $\mathbf{Q}_{SI}^{L,\uparrow}$. Therefore, the depth features can be improved by the complementary information automatically. At the other levels, there is a slight difference in learning the adaptive weights. In specific, at level $l \in \{L-1, L-2, ..., 0\}$, we learn the gating weights $\mathbf{G}_S^l$ using $\mathbf{Q}_{SI}^{l+1,\uparrow}$, rather than $\mathbf{F}_S^l$. Moreover, we feed the concatenated feature $[\mathbf{Q}_{SI}^{l+1,\uparrow}||[\mathbf{F}_S^l||\mathbf{G}_S^l * \mathbf{F}_I^l]]$ into the ResBlock at $l \in \{L-1, L-2, ..., 1\}$ or one convolutional layer at $l = 0$ to get the fused feature. Due to the symmetry of the structure, a similar procedure is employed in the image branch. To illustrate the difference between the proposed fusion strategy and the existing ones, *e.g.,* direct fusion and direct attention fusion, we provide the visual and quantitative comparisons among them in Figure 5 and the ablation study, respectively.

3.5. Branch Integration

By applying the proposed symmetric gated fusion modules, we obtain two sets of features, one from the image branch and the other from the depth branch. Here, we consider two methods, *i.e.,* end-integration and feature-integration, to integrate them together and then obtain the final prediction result.

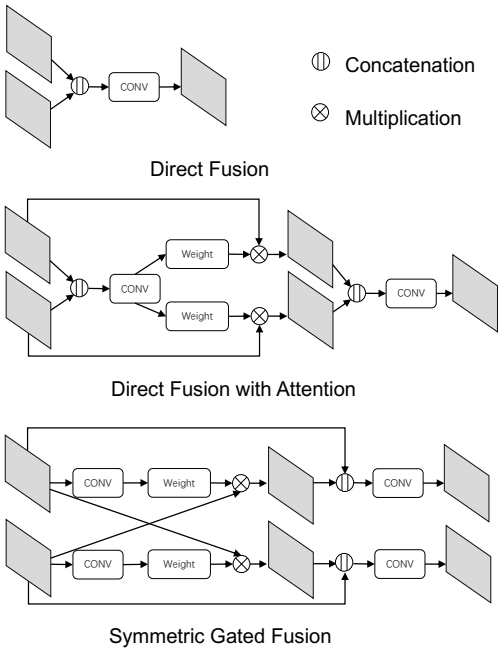


Figure 5: Different fusion strategies. Note that, in implementation, we consider the features in both encoder and decoder.

3.5.1 End-integration

For each branch, we can predict a dense depth map, *i.e.,* $\hat{\mathbf{Y}}_S, \hat{\mathbf{Y}}_I \in \mathbb{R}^{H \times W}$. Since the two branches focus on different information, the reliability of the two predictions varies across the whole image plane. To integrate them adaptively, following [21, 30], we further predict two confidence maps $\mathbf{C}_S, \mathbf{C}_I \in \mathbb{R}^{H \times W}$, which indicate the reliability of the predictions. Therefore, the final dense depth map can be obtained as follows:

$$\hat{\mathbf{Y}} = \frac{exp(\mathbf{C}_S) * \hat{\mathbf{Y}}_S + exp(\mathbf{C}_I) * \hat{\mathbf{Y}}_I}{exp(\mathbf{C}_S) + exp(\mathbf{C}_I)}, \tag{6}$$

where $*$ represents the element-wise multiplication.

3.5.2 Feature-integration

Apart from the integration in the end, we can also combine the features extracted by the two branches. In specific, as shown in Figure 6, we fuse the intermediate features $\mathbf{Q}_{SI}$ and $\mathbf{Q}_{IS}$ through several convolutional operations to obtain $\mathbf{Q}_F$ progressively, and lastly obtain the final prediction $\hat{\mathbf{Y}}$ by applying one convolutional operation on the final integrated features.

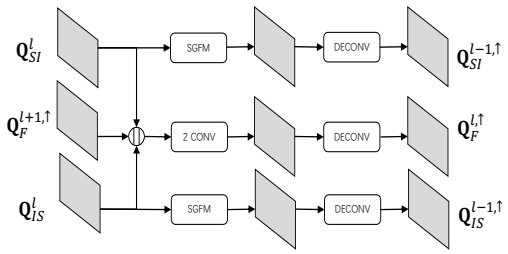


Figure 6: Feature-integration. Note that, we ignore some inputs of the SGFM for simplicity.

3.6. Loss Function

The network is mainly driven by a masked mean squared error (MSE) loss between the ground truth semi-dense depth map $\mathbf{Y}$ and the prediction $\hat{\mathbf{Y}}$, which is defined as:

$$\mathcal{L}_{mse}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N_p} \sum_{i,j} \mathbb{I}(\mathbf{Y}^{i,j} > 0)(\mathbf{Y}^{i,j} - \hat{\mathbf{Y}}^{i,j})^2, \quad (7)$$

where $\mathbb{I}(\cdot)$ denotes the indication function, and N_p represents the number of labeled pixels. In addition, similar to previous works [45], we also apply an edge-aware smoothness loss to encourage depths to preserve spatial continuity:

$$\mathcal{L}_{sm}(\hat{\mathbf{Y}}; \mathbf{X}_I) = \frac{1}{N_s} ||\nabla \hat{\mathbf{Y}}||_1 e^{-||\nabla \mathbf{X}_I||_1}, \quad (8)$$

where N_s denotes the number of pixels in the whole image space, and ∇ represents first derivative along spatial directions. Finally, the full objective is:

$$\begin{aligned} \mathcal{L}(\hat{\mathbf{Y}}, \hat{\mathbf{Y}}_S, \hat{\mathbf{Y}}_I, \mathbf{Y}; \mathbf{X}_I) = & \mathcal{L}_{mse}(\hat{\mathbf{Y}}, \mathbf{Y}) + \\ & \gamma_1 \mathcal{L}_{mse}(\hat{\mathbf{Y}}_S, \mathbf{Y}) + \\ & \gamma_1 \mathcal{L}_{mse}(\hat{\mathbf{Y}}_I, \mathbf{Y}) + \\ & \gamma_2 \mathcal{L}_{sm}(\hat{\mathbf{Y}}; \mathbf{X}_I), \end{aligned} \quad (9)$$

where γ_1 and γ_2 are the trade-off factors, and are set to 0.5 and 0.01 in our experiments, respectively.

4. Experiments

In this section, we first introduce the datasets [4, 25] used in our experiments, and the implementation details. Then we evaluate our method by making comparisons against state-of-the-art methods. Finally, we conduct several ablations to analyze our framework.

4.1. Benchmark Datasets

KITTI Depth Completion Benchmark [4]. It is currently the main benchmark for depth completion. The dataset consists of over $90,000$ frames with the ground truth semi-dense depth map for training and validation, and $1,000$ frames without the ground-truth for test. We train depth completion models on the training set, and then evaluate the performance on the official selected validation and test sets. During training, we crop all training data (images and depth maps, 375×1242) to the size of validation and test data, *i.e.,* 352×1216. For evaluation, we adopt the official error metrics: root mean squared error (RMSE in mm, main metric for ranking), mean absolute error (MAE in mm), root mean squared error of the inverse depth (iRMSE in $1/km$), and mean absolute error of the inverse depth (iMAE in $1/km$).

NYU-v2 [25]. This dataset consists of RGB and depth images collected from 464 different indoor scenes. According to the official data split strategy, 249 scenes are used for training, and 654 labeled images are selected for evaluating the final performance [46, 55]. In our experiments, we sample around $48k$ images with annotations from the training set for training. Adopting similar experimental setting as [10, 28], we firstly down-sample all images to half and center-crop them to 304×228, and then sample 500 sparse LiDAR points from the provided dense depth map randomly as the sparse depth data. We exploit root mean square error (RMSE in $meter$), mean absolute relative error (REL in $meter$), and the percentage of relative errors inside a certain threshold (δ_t, $t \in \{1.25, 1.25^2, 1.25^3\}$) as evaluation metrics.

4.2. Implementation Details

Graph Construction. For KITTI dataset, we build the graphs at three scales with 10000, 5000, and 2500 observed pixels randomly sampled from the downsampled sparse depth maps, respectively, and we calculate 6 nearest neighbours for each node. For NYU-v2, we randomly sample 250, 125, and 60 points, respectively. Note that, we can create the graphs using either the 3D coordinates (*e.g.,* camera coordinates) or the 2D coordinates (*e.g.,* pixel coordinates). Here, we use the 3D coordinates, and we will study the differences in ablation studies.

Architecture Details. At each level of the encoder, we employ two CGPMs, and in the decoder, two ResBlocks are utilized in the symmetric gated fusion module at each scale. The feature channels in the modules are set to 64. Our final results are obtained using the feature-integration, and in this case, we use two convolutional layers, each with 64 output channels at each scale.

Training Details. We implement our depth completion framework in *PyTorch*. In specific, we optimize our network with the momentum of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the initial learning rate of $\alpha = 0.0005$ using the ADAM solver [79]. The model is trained for around 40 epochs with a batch size of 8, and the learning rate is delayed by 0.5 every 10 epochs during training.

Table 1: Quantitative results on the test set of KITTI depth completion benchmark [4], ranked by **RMSE**. The methods ranking first, second, and third are marked by the red, blue, and cyan, respectively. Our method performs better than most of previous methods, and yields close performance to CSPN++ [34] and NLSPN [38] with a much smaller model size.

Method	PARAM (M)	**RMSE**	MAE	iRMSE	iMAE
SparseConvs [9]	-	1601.33	481.27	4.94	1.78
MorphNet [81]	-	1045.45	310.49	3.84	1.57
CSPN [28]	17.41	1019.64	279.46	2.93	1.15
Spade-RGBsD [11]	∼5.3	917.64	234.81	2.17	0.95
HMSNet [19]	-	841.78	253.47	2.73	1.13
DDP [37]	18.8	832.94	203.96	2.10	0.85
NConv-CNN-L2 [20]	0.36	829.98	233.26	2.60	1.03
Sparse2Dense [31]	26.10	814.73	249.95	2.80	1.21
PwP [22]	28.99	777.05	235.17	2.42	1.13
Certainty [30]	2.55	772.87	215.02	2.19	0.93
DeepLiDAR [21]	53.44	758.38	226.05	2.56	1.15
UberATG-FuseNet [15]	1.89	752.88	221.19	2.34	1.14
CSPN++ [34]	∼26	743.69	209.28	2.07	0.90
NLSPN [38]	25.84	741.68	199.59	1.99	0.84
ACMNet	4.9	744.91	206.09	2.08	0.90

4.3. Comparison against the State-of-the-art

KITTI Dataset. In Table 1, we report the number of parameters as well as the performance of our approach and previous peer-reviewed works on KITTI depth completion benchmark. Note that, some of the existing approaches employ additional data during training. For example, DeepLiDAR [21] renders $50K$ training samples using an open urban driving simulator to train the surface normal prediction network, and Certainty [30] utilizes a pre-trained sematic segmentation model on Cityscapes [80] as network initialization, which can provide high-level semantic information for depth completion. In contrast to these approaches, we train our network from scratch without any additional data. Nevertheless, our approach obtains a convincible improvement over most of the previous methods. In comparison to the latest works, *i.e.,* CSPN++ [34] and NLSPN [38], our model achieves very close performance, but our model has fewer parameters. Specifically, the RMSE errors of NLSPN and CSPN++ are $3mm$ and $1mm$ less than ours, respectively, but the number of their parameters is around four times larger than ours.

Figure 7 shows some qualitative results of ACMNet and other four state-of-the-art methods [21, 30, 22, 31]. Benefiting from our proposed co-attention guided graph propagation and symmetric gated fusion strategy, which exploit observed pixels' information and capture the heterogeneity of the two modalities efficiently, ACMNet is capable of yielding high-performing dense depth map, preserving more details over boundary regions (*e.g.,* the 2*nd* and 3*rd* examples), and performing better on the tiny/thin objects (the 1*st* example).

NYU-v2 Dataset. As shown in Table 2, most of latest works have close performance on this dataset. Our method performs better than almost all of methods except NLSPN [38], but as stated above the number of our model's parameters is far less than it.

Table 2: Quantitative results on NYU-v2 [25] with the setting of 500 sparse depth samples. RMSE, REL: lower better; δ_t: higher better.

Method	RMSE	REL	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
TGV [5]	0.635	0.123	81.9	93.0	96.8
Bilateral [25]	0.479	0.084	92.4	97.6	98.9
Zhang *et al.* [8]	0.228	0.042	97.1	99.3	99.7
Ma *et al.* [10]	0.204	0.043	97.8	99.6	99.9
CSPN [28]	0.117	0.016	99.2	**99.9**	**100.0**
DeepLiDAR [21]	0.115	0.022	99.3	**99.9**	**100.0**
Xu *et al.* [22]	0.112	0.018	99.5	**99.9**	**100.0**
NLSPN [38]	**0.092**	**0.012**	**99.6**	**99.9**	**100.0**
ACMNet	0.105	0.015	99.4	**99.9**	**100.0**

Table 3: Quantitative results on KITTI validation set [4] for ablation study on Graph Propagation. Noticeable improvements gained by +GP demonstrate the effectiveness of our proposed graph propagation module.

Method	RMSE	MAE	iRMSE	iMAE
Baseline	815.61	224.43	2.59	1.02
+GP	806.87	220.97	2.42	0.97
+GP/D	810.85	224.64	2.45	0.99
+GP/W	809.09	221.44	2.42	0.97
+SG	796.79	219.86	2.39	0.97
+GP+SG	789.72	216.65	2.32	0.96
+GP/D+SG	792.49	215.14	2.33	0.95
+GP/W+SG	790.75	217.34	2.39	0.97

Table 4: Investigation for different fusion strategies. DF: direct fusion; DAF: direct fusion with attention mechanism; SG: our proposed adaptive symmetric gated fusion strategy.

Method	RMSE	MAE	iRMSE	iMAE
DF	815.61	224.43	2.59	1.02
DAF	807.35	224.70	2.46	1.00
SG	796.79	219.86	2.39	0.97
GP+DF	807.49	218.74	2.39	0.96
GP+DAF	804.69	221.09	2.44	0.99
GP+SG	789.72	216.65	2.32	0.96

4.4. Ablation Study

Here, we conduct comprehensive ablation studies on KITTI selected validation dataset to verify the effectiveness of our proposed components. In following experiments, we set the channels of intermediate layers in networks to 32 to speed up model training. Unless otherwise specified, we exploit the end-integration in most cases.

The effectiveness of the graph propagation. We first demonstrate the effectiveness of the proposed co-attention

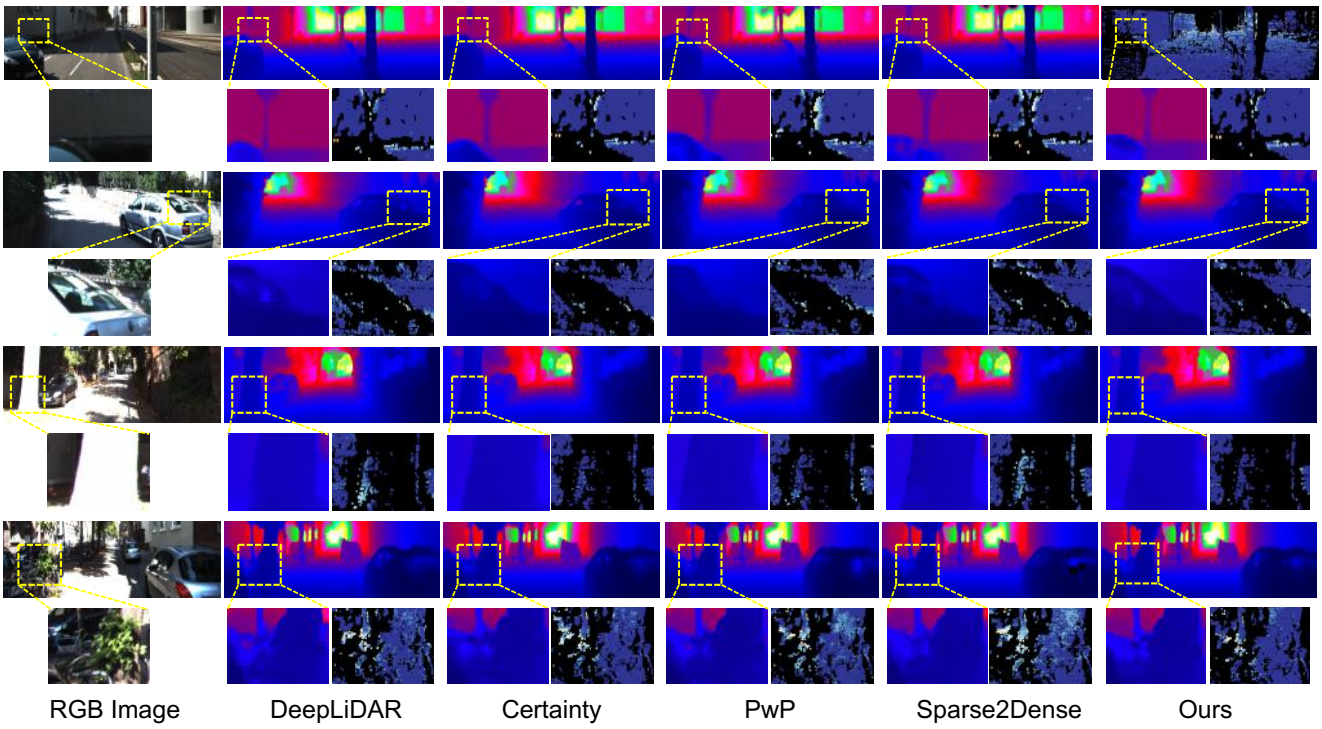


Figure 7: Qualitative comparison of our method against four state-of-the-art approaches on KITTI test set [4]. Left to right: RGB image, results of DeepLiDAR [21], Certainty [30], PwP [22], Sparse2dense [31], and ACMNet, respectively. For better comparison, we show color images, dense predictions, and zoom-in views of details and error maps (darker, better). Best viewed in color.

guided graph propagation by comparing the performance in four cases, *i.e.,* (1) Baseline: no propagation used in the encoder and direct fusion in the decoder; (2) +GP: graph propagation in the encoder and direct fusion in the decoder; (3) +SG: no propagation in the encoder and symmetric gated fusion in the decoder; (4) +GP+SG: our whole model with the end-integration. As shown in Table 3, +GP and +GP+SG outperform Baseline and +SG, respectively, which demonstrates that the proposed graph propagation module better captures the spatial contextual information from sparse LiDAR data.

Furthermore, we carry out four additional experiments to analyze in which stage, such as the encoder (*i.e.,* +GP and +GP+SG), decoder (referred as +GP/D and +GP/D+SG), or whole network (referred as +GP/W and +GP/W+SG), the graph propagation module performs better. As shown in Table 3, the comparisons (+GP *v.s.* +GP/D, and +GP+SG *v.s.* +GP/D+SG) indicate that applying the propagation module in the feature extraction stage is more effective in modeling the contextual information. Additionally, we can also observe that compared to +GP (+GP+SG), +GP/W (+GP/W+SG) causes some performance drop. This might be because in the decoder the structure of the observed pixels is not well-preserved after several operations in the encoder.

The effectiveness of the symmetric gated fusion. To verify that the proposed symmetric gated fusion strategy performs better than direct fusion, *e.g.,* concatenation with or without attention (referred as DAF and DF, respectively), we compare six models, *i.e.,* DF (namely Baseline), DAF, SG, GP+DF (namely Baseline+GP in Table 3), GP+DAF, and GP+SG. As shown in Table 4, SG outperforms both DAF and DF, demonstrating that the proposed symmetric gated fusion strategy is capable of combining the multi-modal information more effectively. Moreover, the comparisons between GP+SG, GP+DAF, and GP+DF can further support this conclusion.

Analysis of graph construction. Here, we investigate the impacts of three factors involved in constructing graphs. Note that, we conduct the following experiments using our final model with the end-integration. We report the results in Table 5.

Firstly, since we aim at capturing more observed multi-modal information to enhance the features of unobserved pixels by finding their spatial neighbours, it is interesting to explore the selection of the coordinate system, *i.e.,* pixel

Table 5: Ablation study on the coordinate system and the number of nearest neighbours and sampled points.

Graph	RMSE	MAE	iRMSE	iMAE
10K_2D_6NN	792.56	216.31	2.34	0.95
10K_3D_6NN	789.72	216.65	2.32	0.96
10K_3D_3NN	792.13	216.64	2.35	0.96
10K_3D_9NN	795.09	216.57	2.37	0.96
08K_3D_6NN	794.59	216.64	2.36	0.95
12K_3D_6NN	793.61	215.81	2.34	0.95

coordinate system or camera coordinate system. In specific, for a set of observed pixels, we can construct a graph according to their 2D coordinates $\{(u_i, v_i)\}_{i=0}^{n-1}$ directly or 3D coordinates $\{(x_i, y_i, z_i)\}_{i=0}^{n-1}$, which are obtained according to Eq. 10, where f_x, f_y, c_x, c_y denote the camera parameters, and d_i represents the depth value. In Table 5, we compare two models (10K_2D_6NN *v.s.* 10K_3D_6NN), where 6-nearest neighbours algorithm is utilized to construct graphs and $10,000$ points are sampled at the first scale. We can find 10K_3D_6NN slightly outperforms 10K_2D_6NN on the RMSE metric. It is mainly because propagation in the camera (3D) coordinate system can learn the scene's geometric structure.

$$\begin{aligned} z_i &= d_i \\ x_i &= \frac{z_i(u_i - c_x)}{f_x} \\ y_i &= \frac{z_i(v_i - c_y)}{f_y} \end{aligned} \tag{10}$$

Secondly, we discuss the performance of the model under different numbers of nearest neighbours. By setting k (k nearest neighbours) to different values, *i.e.,* $3, 6, 9$, we train three models, *i.e.,* 10K_3D_3NN ($k = 3$), 10K_3D_6NN ($k = 6$), and 10K_3D_9NN ($k = 9$), all of which propagate features in the camera coordinate system. As shown in Table 5, in comparison to 10K_3D_3NN and 10K_3D_6NN, 10K_3D_9NN causes a slight decrease in the performance, it might be because increasing the number of nearest neighbours encourages the model to see unrelated contexts.

Lastly, we study the number of sampled points. In specific, we sample $10,000$, $8,000$, and $12,000$ points at the first scale, respectively, and at the following scales, half of points are sampled from the last scale. From Table 5, we can observe that more or fewer points might degrade the performance on the RMSE metric.

In a nutshell, the selection of coordinate system, the number of nearest neighbours and sampled points might affect the performance, but in most settings, the model performs well.

Analysis of branch integration. In Section 3.5, we introduce two methods for the integration of the two branches.

Table 6: Investigation for the two proposed integration methods.

Method	RMSE	MAE	iRMSE	iMAE
End-Integration	789.72	216.65	2.32	0.96
Feature-Integration	786.89	216.24	2.28	0.96
EI/Depth	802.66	219.88	2.40	0.97
EI/Image	807.34	223.26	2.47	1.00

Here, we analyze their performances. As shown in Table 6, the comparison (RMSE: 786 *v.s.* 789) between Feature-Integration (*abbr.* FI) and End-Integration (*abbr.* EI) shows that integration at the feature level is more powerful than the end in learning the reliability of the two branches.

In addition, we also evaluate the performance of the two branches. Taking the end-integration as an example, we report the performance of EI/Depth fusing the RGB information into the depth, and EI/Image doing the opposite. Although the two branches yield close scores on all metrics, by learning confidence maps to fuse them together, a significant improvement on all metrics is obtained. To understand the two branches deeply, we provide a qualitative example in Figure 8. It can be seen that the depth branch is able to generate dense depth map with higher confidence in most locations, while the image branch performs better in capturing the boundary information. This result also further supports that the two modalities are complementary to each other.

4.5. Generalization Capabilities on Different Levels of Sparsity

To show the generalization capabilities of ACMNet on different levels of sparsity, we evaluate our approach and other three state-of-the-art methods with publicly available code, *i.e.*, Certainty [30], Sparse2dense [31], and NConv-CNN [20], on KITTI selected validation set under different input densities. In specific, we first uniformly sub-sample the raw LiDAR depth by ratios of $0.8, 0.6, 0.4, 0.2, 0.1, 0.05$, and 0.025 to generate sparse depth maps with different densities, and then test pretrained models on the generated sparse depth maps. Note that, all the models are trained on KITTI training set under the original sparsity (sampling ratio of 1.0) but not fine-tuned on the new sparse depth maps. Figure 9 shows that our approach performs better under all input densities in terms of both RMSE and MAE metrics, which demonstrates the impressive generalization capabilities of our approach under different levels of sparsity.

5. Conclusion

In this paper, we have developed an Adaptive Context-Aware Multi-Modal Network (ACMNet) to recover a dense depth map from sparse LiDAR data and dense RGB data.

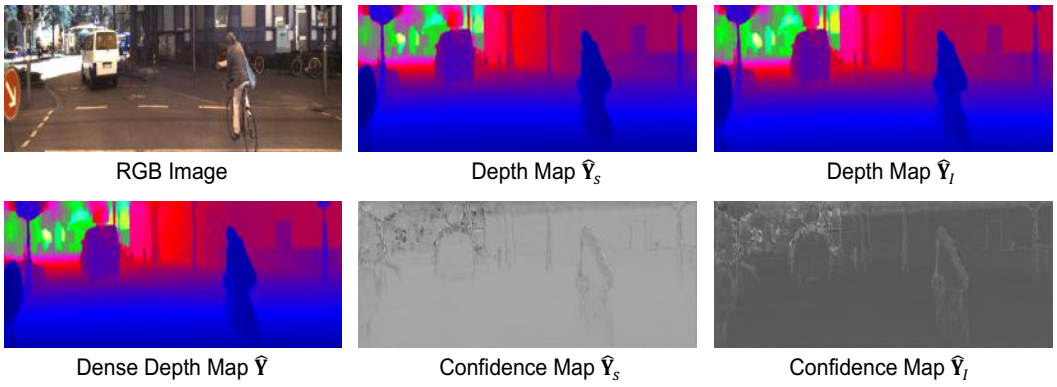
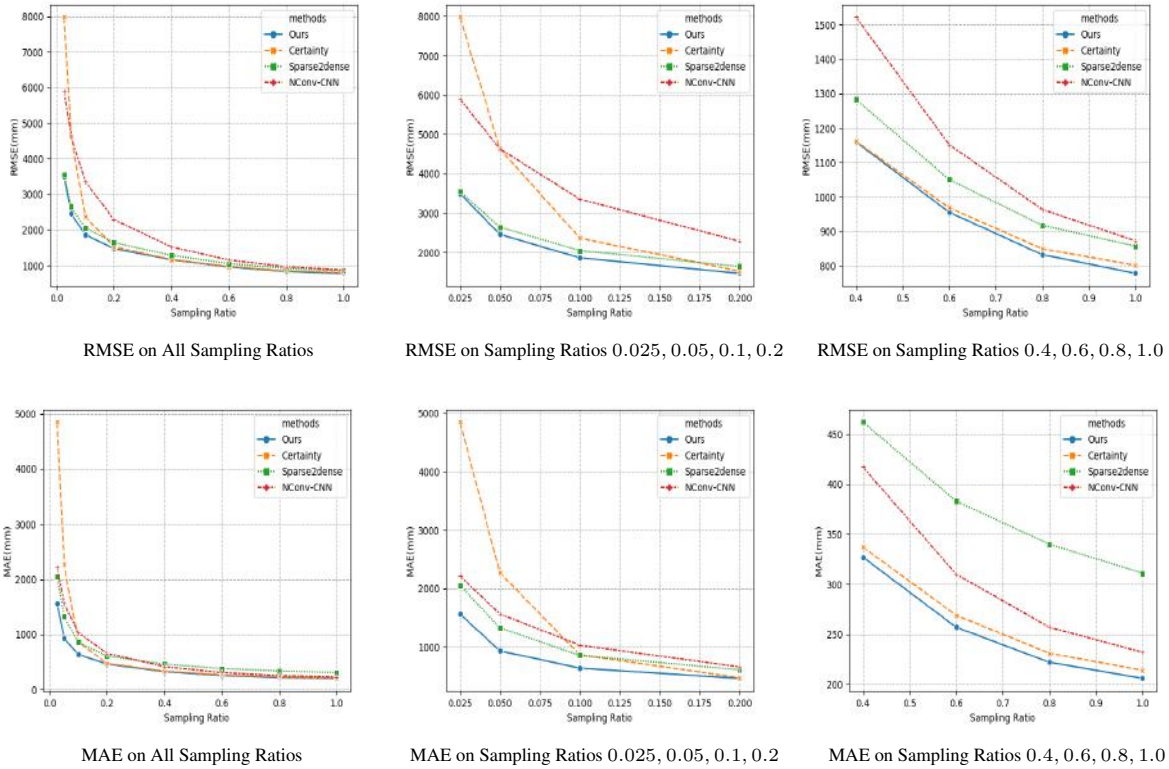


Figure 8: Qualitative example of the end-integration. First row: input image, prediction of EI/Depth and EI/Image, respectively; Second row: final prediction, and confidence maps corresponding to the predictions in the first row. We can find that each branch can capture different information.



RMSE on All Sampling Ratios | RMSE on Sampling Ratios $0.025, 0.05, 0.1, 0.2$ | RMSE on Sampling Ratios $0.4, 0.6, 0.8, 1.0$

MAE on All Sampling Ratios | MAE on Sampling Ratios $0.025, 0.05, 0.1, 0.2$ | MAE on Sampling Ratios $0.4, 0.6, 0.8, 1.0$

Figure 9: Performances under different levels of sparsity. For better comparison, we also show the performances on lower and larger densities separately in the right two figures of each row. In comparison to Certainty [30], Sparse2dense [31], and NConv-CNN [20], ACMNet performs better under all input densities.

The critical issue in depth completion is how to exploit the observed spatial contexts from multi-modal data efficiently. To this end, we apply the co-attention guided graph propagation within multiple graphs constructed from observed pixels, which adaptively extracts multi-scale and multi-modal features and contributes to the feature enhancement for unobserved pixels. Furthermore, to fuse the multi-modal features in an effective way, we propose the symmetric gated fusion strategy, which has the capability of learning the heterogeneity of the two modalities. Finally,

we implement our ACMNet, where a stack of CGPMs are employed in the encoder and SGFMs are used in the decoder. Benefiting from the two new modules, ACMNet is capable of generating high-quality dense depth maps. Our extensive experiments have demonstrated the effectiveness of the network as well as the network components.

References

[1] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," *arXiv preprint arXiv:1901.04780*, 2019. 1

[2] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[3] G. Moon, J. Yong Chang, and K. Mu Lee, "V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2, 7, 8, 9

[5] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 993–1000. 1, 2, 8

[6] D. Herrera, J. Kannala, J. Heikkilä *et al.*, "Depth map inpainting under a second-order smoothness prior," in *Scandinavian Conference on Image Analysis*. Springer, 2013, pp. 555–566. 1, 2

[7] N. Schneider, L. Schneider, P. Pinggera, U. Franke, M. Pollefeys, and C. Stiller, "Semantically guided depth upsampling," in *German Conference on Pattern Recognition*. Springer, 2016, pp. 37–48. 1, 2

[8] Y. Zhang and T. Funkhouser, "Deep depth completion of a single rgb-d image," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 3, 8

[9] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 11–20. 1, 2, 4, 8

[10] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8. 1, 7, 8

[11] M. Jaritz, R. De Charette, E. Wirbel, X. Perrotton, and F. Nashashibi, "Sparse and dense data with cnns: Depth completion and semantic segmentation," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 52–60. 1, 2, 3, 4, 8

[12] S. Imran, Y. Long, X. Liu, and D. Morris, "Depth coefficients for depth completion," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[13] A. Atapour-Abarghouei and T. P. Breckon, "Veritatem dies aperit - temporally consistent depth prediction enabled by a multi-task geometric and semantic scene understanding approach," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 4

[14] X. Cheng, Y. Zhong, Y. Dai, P. Ji, and H. Li, "Noise-aware unsupervised deep lidar-stereo fusion," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[15] Y. Chen, B. Yang, M. Liang, and R. Urtasun, "Learning joint 2d-3d representations for depth completion," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 8

[16] Y. Zhong, C.-Y. Wu, S. You, and U. Neumann, "Deep rgb-d canonical correlation analysis for sparse depth completion," in *Advances in Neural Information Processing Systems*, 2019, pp. 5332–5342. 1

[17] A. Eldesokey, M. Felsberg, K. Holmquist, and M. Persson, "Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2

[18] K. Lu, N. Barnes, S. Anwar, and L. Zheng, "From depth what can you see? depth completion via auxiliary image reconstruction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2

[19] Z. Huang, J. Fan, S. Cheng, S. Yi, X. Wang, and H. Li, "Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion," *IEEE Transactions on Image Processing*, vol. 29, pp. 3429–3441, 2020. 1, 2, 4, 8

[20] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through cnns for guided sparse depth regression," *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1, 2, 4, 8, 10, 11

[21] J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys, "Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3313–3322. 1, 2, 3, 4, 6, 8, 9

[22] Y. Xu, X. Zhu, J. Shi, G. Zhang, H. Bao, and H. Li, "Depth completion from sparse lidar data with depth-normal constraints," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 3, 4, 8, 9

[23] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *arXiv preprint arXiv:1801.07829*, 2018. 1, 3

[24] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Advances In Neural Information Processing Systems*, 2016, pp. 289–297. 2, 5

[25] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*. Springer, 2012, pp. 746–760. 2, 7, 8

[26] J. T. Barron and B. Poole, "The fast bilateral solver," in *European Conference on Computer Vision*. Springer, 2016, pp. 617–632. 2

[27] N. Chodosh, C. Wang, and S. Lucey, "Deep convolutional compressed sensing for lidar depth completion," *arXiv preprint arXiv:1803.08949*, 2018. 2

[28] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 103–119. 2, 3, 7, 8

[29] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning guided convolutional network for depth completion," *arXiv preprint arXiv:1908.01238*, 2019. 2, 3, 4

[30] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy lidar completion with rgb guidance and uncertainty," in *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE, 2019, pp. 1–6. 2, 3, 4, 6, 8, 9, 10, 11

[31] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3288–3295. 2, 4, 8, 9, 10, 11

[32] Y. Yang and S. Soatto, "Conditional prior networks for optical flow," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 271–287. 2, 3

[33] A. Atapour-Abarghouei and T. P. Breckon, "To complete or to estimate, that is the question: A multi-task approach to depth completion and monocular depth estimation," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 183–193. 2

[34] X. Cheng, P. Wang, C. Guan, and R. Yang, "Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion." in *AAAI*, 2020, pp. 10 615–10 622. 2, 3, 8

[35] Y. Liao, L. Huang, Y. Wang, S. Kodagoda, Y. Yu, and Y. Liu, "Parse geometry from a line: Monocular depth estimation with partial laser observation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5059–5066. 2

[36] A. Li, Z. Yuan, Y. Ling, W. Chi, C. Zhang *et al.*, "A multi-scale guided cascade hourglass network for depth completion," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 32–40. 2

[37] Y. Yang, A. Wong, and S. Soatto, "Dense depth posterior (ddp) from single image and sparse range," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3, 8

[38] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. S. Kweon, "Non-local spatial propagation network for depth completion," *arXiv preprint arXiv:2007.10042*, 2020. 3, 8

[39] X. Xiong, H. Xiong, K. Xian, C. Zhao, Z. Cao, and X. Li, "Sparse-to-dense depth completion revisited: Sampling strategy and graph construction," in *European Conference on Computer Vision (ECCV)*, 2020. 3

[40] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018. 3

[41] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in neural information processing systems*, 2006, pp. 1161–1168. 3

[42] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 824–840, 2009. 3

[43] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2002–2011. 3

[44] S. Liu, J. Pan, and M.-H. Yang, "Learning recursive filters for low-level vision via a hybrid neural network," in *European Conference on Computer Vision*. Springer, 2016, pp. 560–576. 3

[45] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 270–279. 3, 7

[46] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374. 3, 7

[47] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858. 3

[48] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision*. Springer, 2016, pp. 740–756. 3

[49] L. He, G. Wang, and Z. Hu, "Learning depth from single images with deep neural network embedding focal length," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4676–4689, 2018. 3

[50] Y. Kim, H. Jung, D. Min, and K. Sohn, "Deep monocular depth estimation via integration of global and local predictions," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4131–4144, 2018. 3

[51] Y. Cao, T. Zhao, K. Xian, C. Shen, Z. Cao, and S. Xu, "Monocular depth estimation with augmented ordinal depth relationships," *IEEE Transactions on Image Processing*, pp. 1–1, 2018. 3

[52] A. Wang, Z. Fang, Y. Gao, S. Tan, S. Wang, S. Ma, and J. Hwang, "Adversarial learning for joint optimization of depth and ego-motion," *IEEE Transactions on Image Processing*, vol. 29, pp. 4130–4142, 2020. 3

[53] H. Yang, P. Chen, K. Chen, C. Lee, and Y. Chen, "Fade: Feature aggregation for depth estimation with multi-view stereo," *IEEE Transactions on Image Processing*, vol. 29, pp. 6590–6600, 2020. 3

[54] Z. Zhang, C. Xu, J. Yang, J. Gao, and Z. Cui, "Progressive hard-mining network for monocular depth estimation," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3691–3702, 2018. 3

[55] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 239–248. 3, 7

[56] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2024–2039, 2016. 3

[57] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658. 3

[58] A. Atapour-Abarghouei and T. P. Breckon, "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2800–2810. 3

[59] S. Zhao, H. Fu, M. Gong, and D. Tao, "Geometry-aware symmetric domain adaptation for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9788–9798. 3

[60] C. Zheng, T.-J. Cham, and J. Cai, "T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 767–783. 3

[61] J. Nath Kundu, P. Krishna Uppala, A. Pahuja, and R. Venkatesh Babu, "Adadepth: Unsupervised content congruent adaptation for depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2656–2665. 3

[62] K. PNVR, H. Zhou, and D. Jacobs, "Sharingan: Combining synthetic and real data for unsupervised geometry estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[63] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009. 3

[64] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7912–7921. 3

[65] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 3

[66] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 296–10 305. 3, 5

[67] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," *arXiv preprint arXiv:1711.04043*, 2017. 3

[68] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11–20. 3

[69] Y. Wu, O. E. F. Bourahla, X. Li, F. Wu, Q. Tian, and X. Zhou, "Adaptive graph representation learning for video person re-identification," *IEEE Transactions on Image Processing*, 2020. 3

[70] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome, "Mutan: Multimodal tucker fusion for visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2612–2620. 3

[71] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576. 3

[72] J. H. Yoo, Y. Kim, J. S. Kim, and J. W. Choi, "3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection," *arXiv preprint arXiv:2004.12636*, 2020. 3

[73] C. Hori, T. Hori, G. Wichern, J. Wang, T.-y. Lee, A. Cherian, and T. K. Marks, "Multimodal attention for fusion of audio and spatiotemporal features for video description." in *CVPR Workshops*, 2018, pp. 2528–2531. 3

[74] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9621–9630. 5

[75] J. Li, B. M. Chen, and G. Hee Lee, "So-net: Self-organizing network for point cloud analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9397–9406. 5

[76] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008. 5

[77] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3. 5

[78] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 5, 6

[79] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 7

[80] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223. 8

[81] M. Dimitrievski, P. Veelaert, and W. Philips, "Learning morphological operators for depth completion," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2018, pp. 450–461. 8