Pattern Recognition 166 (2025) 111645

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Domain adaptive depth completion via spatial-error consistency

Lingyu Xiao [a], Jinhui Wu [a], Junjie Hu [c], Ziyu Li [a], Wankou Yang [a,b,*]

[a] School of Automation, Southeast University, Nanjing 210096, China
[b] Advanced Ocean Institute of Southeast University, Nantong, China
[c] Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518129, China

ARTICLE INFO

Keywords:
Depth completion
Domain adaptation
Uncertainty estimation
Autonomous vehicles

ABSTRACT

In this paper, we introduce a novel training framework designed to address the challenge of unsupervised domain adaptation (UDA) in depth completion. Our framework aims to bridge the gap between lidar and image data by establishing a shared domain, which is a collection of the confidence of the network's prediction. By indirectly adapting the depth network through this common domain, the problem is decomposed into two key tasks: (1) constructing the common domain and (2) adapting the depth network using the common domain. For the construction of the common domain, errors in the network's predictions are modelled as confidence, which serves as supervision for a sub-module called the Depth Completion Plugin (DCPlugin). The purpose of the DCPlugin is to generate the confidence associated with any given dense depth prediction. To adapt the depth network using the common domain, a confidence-aware co-training task is employed, leveraging the confidence map provided by the well-adapted DCPlugin. To assess the effectiveness of our proposed approach, we conduct experiments on multiple depth networks under adaptation scenarios, namely CARLA → KITTI and VKITTI → KITTI. The results demonstrate that our method surpasses other domain adaptation (DA) techniques, achieving state-of-the-art performance. Given the limited existing work in this domain, we provide comprehensive discussions to guide future researchers in this field.

1. Introduction

Recent advancements in deep learning technology have significantly revolutionized algorithm design across multiple research fields, particularly in computer vision and robotics. Depth completion, an essential component among autonomous driving, SFM/SLAM, and 3D reconstruction, also plays an indispensable role.

Although recent state-of-the-art supervised depth completion pipelines [1–6] have achieved impressive results, they require abundant annotations, which are often inaccessible in the real world due to the labour-intensive and occasionally inaccurate process of annotating depth information. While training data with dense annotations in a synthetic world is theoretically infinite and easy to obtain, the main challenge lies in transferring the learned knowledge from the source domain (synthetic world) to the target domain (real world) without relying on any priors. This task falls under the domain of Unsupervised Domain Adaptation (UDA).

However techniques introduced by previous works on semantic segmentation [7–10], object detection [11,12], and classification [13,14] face difficulties when applied to depth completion due to the following perspectives: First, unlike these tasks, the encoder–decoder formulation is not a universal design in depth completion, as exemplified by [1]. Second, depth completion is a multi-modality task, making it tremendously challenging to align features between domains. Moreover, although a recent work [15] introduced a well-designed pipeline for obtaining training data from the Carla simulator [16] (referred as CARLA for convenience since the data used in their work is not officially released as a benchmark) to address the UDA problem, the setup for data acquisition is complicated, hindering its widespread adoption and promotion.

In this paper, to address the aforementioned challenges, we propose a two-stage training framework which aims to indirectly adapt the depth completion network through a more manageable common domain, and it includes: (1) constructing the common domain (Stage I) and (2) adapting the depth network using the common domain (Stage II).

In the context of depth completion, a clear structure/domain-agnostic pattern emerges: ***spatial-error consistency across domains***. Specifically, across different domains, the error predicted by depth completion network escalates incessantly as the distance extends, while it exhibits a pronounced surge in magnitude precisely at the objects' edges, as illustrated in Fig. 1.

* Corresponding author at: School of Automation, Southeast University, Nanjing 210096, China.
E-mail address: wkyang@seu.edu.cn (W. Yang).

https://doi.org/10.1016/j.patcog.2025.111645
Received 11 September 2024; Received in revised form 18 February 2025; Accepted 24 March 2025
Available online 15 April 2025

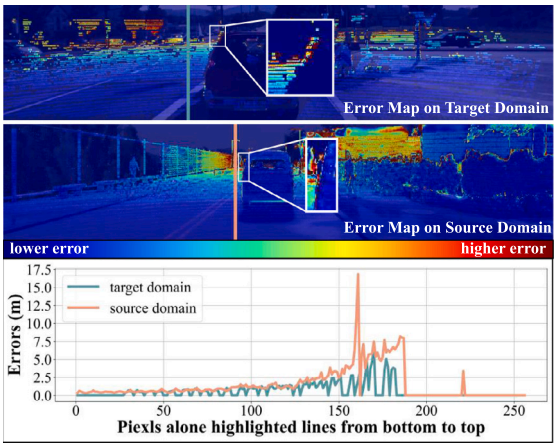


Fig. 1. Our basic structural-agnostic and domain-agnostic observation under depth completion task: ***spatial-error consistency across domains***. We noticed that large errors tend to occur at the edges of objects and distant objects, while smaller errors are observed in the central and close regions. This observation is illustrated in the first and second row images, which show the errors in the predicted depth from the target and source domains, respectively. The last row graph demonstrates the continuous increase in error along two highlighted lines, from bottom to top, which is consistent across both domains and serves as the basis for constructing the common domain.

Building upon this fundamental observation, we approach the construction of the common domain by representing it as the certainty of depth completion network's predictions. To achieve this, we model the confidence through errors and introduce a sub-module called the Depth Completion Plugin (DCPlugin), which generates a confidence map associated with each prediction made by depth completion network. Thus, the DCPlugin serves as a tangible representation of the common domain.

In Stage II of our framework, we incorporate a confidence-aware co-training task that leverages the confidence map provided by the DCPlugin. Specifically, we implement a teacher–student scheme where the pseudo labels generated by the teacher model are guided by the DCPlugin to identify its trustworthy regions. Furthermore, to compensate for the absence of *aleatoric* uncertainty during the construction of the common domain, we propose a confidence map refinement scheme (CMR) in this stage. Full pipeline is demonstrated in Fig. 3.

The primary differences between our method and previous approaches are depicted in Fig. 2. Methods (a) and (b) aim to minimize the domain gap by addressing disparities in the training data, specifically using RGB data or a combination of RGB and point cloud data between the source domain and target domain. This often results in higher prediction errors or a lack of geometric consistency. Our method improves upon (a) and (b) by minimizing the domain gap through a self-constructed common domain, which is based on the confidence map of the DC network's predictions. This approach directly influences the supervision signal rather than modifying the training data, thereby achieving both geometric consistency and lower prediction errors. The code and models will be made available in the future.

The main contributions of this paper are summarized as:

- We are inspired by the spatial-error consistency under depth completion and propose a two-stage training framework. Moreover, this property may be applicable to other dense prediction tasks, we hope this would inspire researchers from other fields.
- We introduce a structure-agnostic sub-module to serve as a tangible representation of the common domain, through which we indirectly perform domain adaption on the depth completion network.
- We propose a confidence map refinement scheme (CMR) to minimize the domain gap within the common domain.
- Through experiments conducted on adaptation scenarios CARLA→KITTI and VKITTI→KITTI using multiple depth completion networks, we achieve state-of-the-art performance when compared to other DA methods.

2. Related works

2.1. Depth completion

The original task of depth completion involves generating dense lidar output from sparse lidar input [17]. However, the precision of the dense prediction is directly influenced by the sparsity of the lidar input [18]. To address this limitation, recent state-of-the-art methods [1–6] have introduced the fusion of RGB information with the sparse lidar input to compensate for the missing point cloud data. Single branch-based methods, such as [2], directly adapt ResNet [19] into a UNet-like [20] structure. Following methods like [21] employ spatial propagation networks to refine depth. Multi-branch-based approaches, including [3–6], concatenate RGB and point cloud data separately. Specifically, the Bilateral propagation Network (BP-Net) [22] utilizes an early-stage bilateral propagation phase to generate an initial dense depth map, enhancing the subsequent multi-modal fusion and depth refinement stages. DFU [23] employs confidence-aware guidance with adaptive receptive fields to prevent feature loss commonly seen in encoder–decoder networks, significantly improving depth estimation accuracy and generalizability. TPVD [24] leverages both 2D and 3D feature interactions through multi-view geometric fusion, yielding superior performance. However, supervised methods often require massive amounts of labelled data for training, which is labour-intensive in the real world. For Self-supervised methods, as discussed in [2,4,25–28], they leverage additional available priors, such as Visual-Inertial Odometry (VIO) [26], geometry transformation [2], pre-trained encoders on synthetic datasets [29], learnable camera intrinsics [25], or scale-consistent absolute depth values [28], to mimic supervision. However, self-supervised methods only provide sparse supervision by maintaining depth consistency on valid depth points, which leads to underperformance compared to supervised methods [18]. Furthermore, in real-world applications, calibrated onboard systems may experience slight sensor drifting during vehicle operation [30], resulting in potentially incorrect calibration information present in both training and validation data.

2.2. Domain adaptation

Generally, to address DA problems two major approaches have been proposed: (1) Learning domain-invariant features at the feature level [10,31], and (2) Learning pixel-level representations from the source domain to the target domain [7]. In the context of depth estimation, several approaches have been proposed following these two pipelines. [32] utilizes style transfer to learn pixel-level representations from the source to the target domain. [33] aligns feature-level consistency through adversarial learning. [34,35] combine both approaches for improved performance. However, these methods often require additional training for image translation networks and precise feature extraction. The former can be overwhelming, while the latter may not be applicable to multi-branch based methods.

Furthermore, [36] proposes pseudo-labelling methods for domain adaptation but overlooks the information behind unreliable predictions in the 2D-based phase. For depth completion, only [15] discusses the UDA problem by designing a sophisticated pipeline to obtain lidar input with artefacts, mimicking the depth collected by KITTI. However, it still lags far behind the performance of the self-supervised method [25].

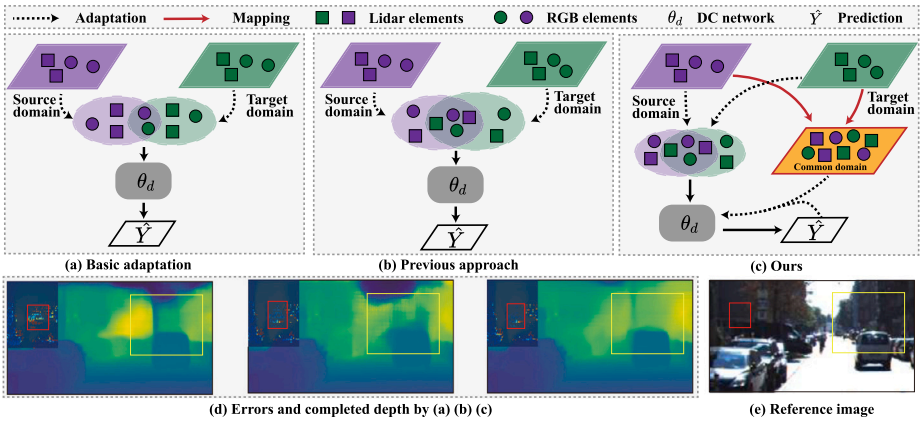


Fig. 2. Illustration of three adaptation paradigms in depth completion. Closer areas are depicted with darker colours, while distant areas are shown with lighter colours. (a) only perform adaption on image data and (b) [15] perform on both lidar and image. (c) our approach proposes to map data from the source domain and target domain into a more manageable common domain, and indirectly adapt depth completion network through the common domain. (a) maintains geometry consistency (highlighted by the yellow rectangle) but yields unsatisfactory results (highlighted by the red rectangle). (b) achieves better results but sacrifices geometry consistency. In contrast, our method demonstrates promising results while retaining geometry consistency. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

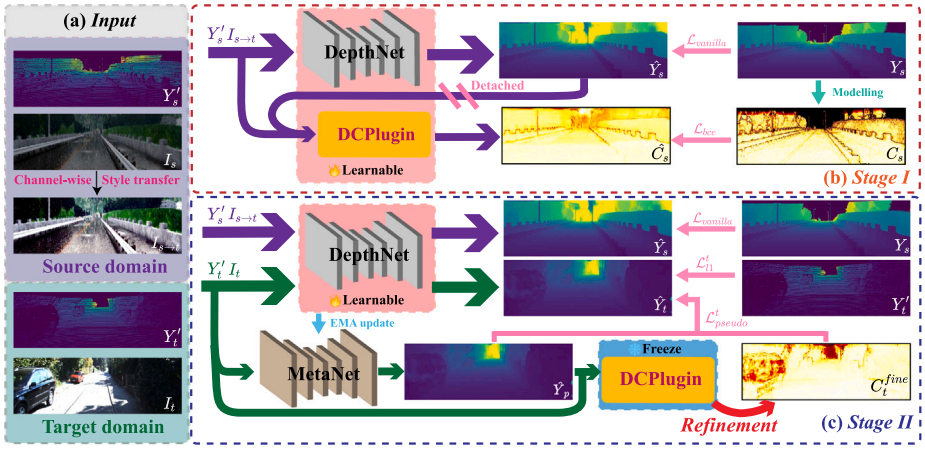


Fig. 3. The full pipeline of our training scheme. DepthNet is depth completion network, a detailed description is presented in the main text.

2.3. Uncertainty estimation

To estimate uncertainty/certainty, Bayesian deep learning provides a foundational framework [37]. Within this category, there are two branches: *epistemic* and *aleatoric*. *Epistemic* focuses on the uncertainty caused by the Deep Neural Network (DNN), while *aleatoric* captures noise within the data. A comprehensive discussion of their pros and cons can be found in [38]. Various methods, such as dropout [39] and ensemble [40], are widely used to discover uncertainty within DNNs. In the context of depth estimation, [41] obtains uncertainty through the variants of two input sources, [38] proposes a self-training scheme to obtain variance from a teacher model, and [42] introduces gradients as a post hoc uncertainty estimation approach. However, the modification of depth completion network and the specific training pipeline prevent their application in our approach, where we aim for a structural-agnostic approach with a simple pipeline.

3. The proposed method

The full pipeline is demonstrated in Fig. 3. Our training scheme consists of two stages: Stage I and Stage II. In Stage I, only transferred source domain data is utilized to perform supervised training on depth completion network. Additionally, we use Eq. (16) to model the ground truth confidence map using depth completion network predictions and ground truth depth from the source domain data to provide supervision for DCPlugin. In Stage II, we implement a teacher–student scheme to perform co-training, where depth completion network acts as the student and a structurally identical MetaNet as the teacher.

3.1. DCPlugin

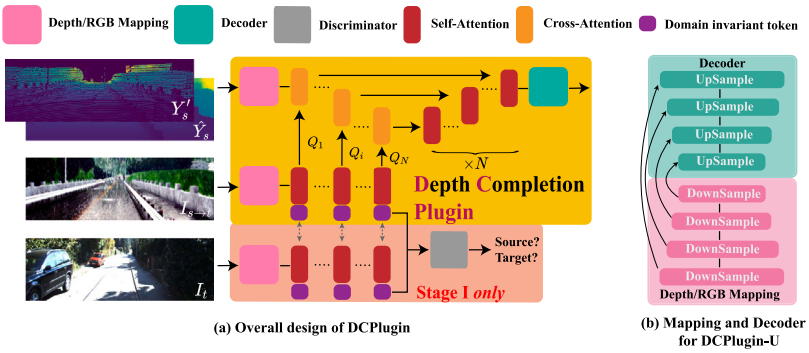
Motivation. As discussed in Section 2.3, estimating uncertainty typically requires a specific training scheme or modifications to depth completion network itself, which poses challenges under our structural-agnostic assumption. However, we have devised an alternative approach to address this issue. We leverage a DCPlugin to capture the inherent uncertainty within depth completion network.

Furthermore, in Stage II, we incorporate a confidence refinement scheme to effectively capture the noise present in the data.

Objective. The objective of DCPlugin is learning a confidence map $\hat{C}$ (we use $\hat{}$ to denote predicted one) given any dense depth prediction $\hat{Y}$. Physically speaking, the confidence map C of $\hat{Y}$ can be represented by its variance Σ, written as $C = f(\Sigma)$, f is the confidence modelling function will be discussed in Section 3.2. Therefore after applying Bayes' theorem, we would write the objective as:

$$P(\Sigma|\hat{Y},\theta_p) \approx P(\Sigma|X,\theta_p) = \frac{P(\Sigma|\hat{Y},X,\theta_p)P(\hat{Y}|X,\theta_p)}{P(\hat{Y}|\Sigma,X,\theta_p)} \tag{1}$$

Here, X represents the input sparse depth Y' and image I, $\hat{Y}$ represents the output dense depth by depth completion network $\mathcal{N}(X)$ and θ_p is



(a) Overall design of DCPlugin

(b) Mapping and Decoder for DCPlugin-U

Fig. 4. Illustration of DCPlugin. We provide the overall design for DCPlugin (a) and a detailed design for DCPlugin-U (b). Both depth and image are first mapped into the feature space, either through patch embedding or downsampling. After several rounds of context aggregation using attention modules, a decoder computes the confidence map. To bridge the domain gap in the confidence map, an additional adversarial learning scheme is introduced in the image branch, this scheme aims to extract domain-invariant features that serve as queries for the depth branch.

parameters in DCPlugin. Since $\hat{Y}$ is calculated from X by $\hat{Y} = \mathcal{N}(X)$, $\hat{Y}$ and X are not mutually independent. Therefore the equation can be rewritten as:

$$P(\Sigma|X,\theta_p) = P(\Sigma|\hat{Y},X,\theta_p), \tag{2}$$

thus the objective can be simplified as:

$$\underset{\theta_p}{\text{argmax}}\ P(\Sigma|\hat{Y},X,\theta_p). \tag{3}$$

Structure. Recent advancements in transformer technology [1,43] have positioned it as a powerful tool for multiple tasks. Our design for DCPlugin is illustrated in Fig. 4(a). It consists of Depth/RGB Mapping, Cross-Attention (CA), Self-Attention (SA), and Decoder. Empirically, we perform adversarial learning on the domain-invariant to stabilize the training. We propose two types of DCPlugin: the 'E' type and the 'U' type. The main difference between the 'E' type and the 'U' type lies in the Depth/RGB Mapping and Decoder. The 'E' type uses patch embedding based on [43] for mapping and a linear decoder. On the other hand, the 'U' type employs UNet-like downsampling and upsampling for mapping and decoding illustrated in Fig. 4(b), which preserves more details but requires more parameters compared to the 'E' type. The description below is based on the 'U' type.

Depth/RGB Mapping & Decoder. The input for the DCPlugin is constituted of a depth input $\mathbf{X}_D$, and an image input $\mathbf{X}_I$.

For depth input $\mathbf{X}_D = [\hat{Y}, Y] \in \mathbb{R}^{2\times H\times W}$, it will be downsampled four times and finally fed into CA and SA to aggregate the downsampled RGB information (the image input is $\mathbf{X}_I \in \mathbb{R}^{3\times H\times W}$). Meanwhile, for each layer, the low-level features are reserved and fed directly into the decoder.

Mathematically, the process of Mapping can be represented as:

$$\begin{aligned}\mathbf{x}^D &= \mathcal{M}_D(\mathbf{X}_D) \in \mathbb{R}^{L\times C},\\ \mathbf{x}^I &= \mathcal{M}_I(\mathbf{X}_I) \in \mathbb{R}^{L\times C}.\end{aligned} \tag{4}$$

$L = \frac{H\times W}{16^2}$ is the length of the token that is fed to SA or CA. C represents the number of channels in the feature extracted by $\mathcal{M}_D$ or $\mathcal{M}_I$, which correspond to the depth and image mappings, respectively.

Self-attention & Cross-attention. SA follows the design of transformer layer [43], which consists of a multi-headed self-attention (MSA) block, MLP block and layer norm (LN).

$$\begin{aligned}\mathbf{x}_i &= \text{SA}(\mathbf{x}_{i-1}) = \text{MLP}(\text{LN}(\mathbf{a}_{i-1})) + \mathbf{a}_{i-1},\\ \mathbf{a}_{i-1} &= \text{MSA}(\text{LN}(\mathbf{x}_{i-1})) + \mathbf{x}_{i-1},\end{aligned} \tag{5}$$

i is the layer index of SA, ranging from 1 to N. Supposing for an input $\mathbf{X} \in \mathbb{R}^{L\times C}$, MSA is formulated as:

$$\text{MSA}(\mathbf{X}) = \text{softmax}(\frac{\mathbf{X}\mathbf{X}^T}{\sqrt{C}})\mathbf{X}. \tag{6}$$

Similarly, given an input $\mathbf{x}_1$, CA is designed via the following equations:

$$\begin{aligned}\mathbf{x}_i &= \text{CA}(\mathbf{x}_{i-1},\mathbf{Q}_{i-1}) = \text{MLP}(\text{LN}(\mathbf{a}_{i-1})) + \mathbf{a}_{i-1},\\ \mathbf{a}_{i-1} &= \text{MCA}(\text{LN}(\mathbf{x}_{i-1}),\mathbf{Q}_{i-1}) + \mathbf{x}_{i-1},\end{aligned} \tag{7}$$

Where the multi-headed cross-attention (MCA) block is formulated as:

$$\text{MCA}(\mathbf{X},\mathbf{Q}) = \text{softmax}(\frac{\mathbf{X}\mathbf{Q}^T}{\sqrt{C}})\mathbf{X}. \tag{8}$$

Domain-invariant Token. Noting $L = \frac{H\times W}{16^2}$ is the length of the token that is fed to SA or CA and K is the number of channels in the extracted feature. Given image token $\mathbf{x}_1^I \in \mathbb{R}^{(L+1)\times K} = [\mathbf{x}^I, \mathbf{t}^I]$ and lidar token $\mathbf{x}_1^D \in \mathbb{R}^{(L+1)\times K} = [\mathbf{x}^D, \mathbf{t}^D]$ from Depth/RGB Mapping, we can obtain $\mathbf{x}_{2N}^D \in \mathbb{R}^{(L+1)\times K}$ from SA and CA using the following equations:

$$\begin{aligned}&\mathbf{x}_i^I = \text{SA}(\mathbf{x}_{i-1}^I),\ 1 \leqslant i \leqslant N,\\ &\mathbf{x}_i^D = \text{CA}(\mathbf{x}_{i-1}^D, \mathbf{x}_{i-1}^I),\ 1 \leqslant i \leqslant N,\\ &\mathbf{x}_i^D = \text{SA}(\mathbf{x}_{i-1}^D),\ i = N+1,\\ &\mathbf{x}_i^D = \text{SA}(\mathbf{x}_{i-1}^D + \mathbf{x}_{2N-i}^D),\ N+1 < i \leqslant 2N.\end{aligned} \tag{9}$$

Here, $\mathbf{t}^D$ and $\mathbf{t}^I$ represent domain-invariant tokens, with the latter being used for adversarial learning in Stage I, and the former used for fusing domain-invariant features learned through adversarial learning. Both $\mathbf{t}^D$ and $\mathbf{t}^I$ are learnable during the training phase.

Since $\mathbf{x}_{2N}^D \in \mathbb{R}^{(L+1)\times K}$, to utilize $\mathbf{x}_{2N}^D$ for the decoder, we detach the first L tokens from $\mathbf{x}_{2N}^D$, which contain both the original information and the fused domain-invariant representation. The detached tokens are then reshaped into the downsampled shape $\mathbf{x} \in \mathbb{R}^{C\times\frac{H}{16}\times\frac{W}{16}}$ and upsampled to $H \times W$ using either direct or progressive bilinear upsampling.

3.2. Stage I: Constructing the common domain

Objective. In the process of constructing the common domain, two tasks should be addressed: learning basic knowledge from the source domain for depth completion network and learning an attached submodule called DCPlugin, which can estimate pixel-level confidence given any dense depth and associated input. The latter task is more challenging than the former, as pixel-level confidence is highly coupled with depth completion network's output. Therefore, a discussion on confidence modelling is necessary.

Modelling Confidence. The objective of optimizing depth completion network is maximizing the posterior probability:

$$P(\hat{Y}|X,\theta_d), \tag{10}$$

θ_d are the parameters in depth completion network. We follow [44] to introduce Σ, together with Bayes' theorem, Eq. (10) can be rewritten as:

$$P(\hat{Y},\Sigma|X,\theta_d) = P(\Sigma|X,\theta_d)P(\hat{Y}|\Sigma,X,\theta_d). \tag{11}$$

Based on Jeffrey's prior [45], the former likelihood can be approximated as $P(\Sigma|X,\theta_d) \approx \frac{1}{\Sigma}$ since the input depth is sparse. For the latter likelihood, we assume that it follows the Laplace distribution as described in [46], resulting in:

$$P(\hat{Y}|\Sigma,X) \approx \frac{1}{2\Sigma}\exp(-\frac{|Y-\hat{Y}|}{\Sigma}), \tag{12}$$

Y is ground truth dense depth, $\hat{Y}$ is the predicted dense depth that is determined by depth completion network's parameters θ_d.

Based on the above deduction, the object of maximizing Eq. (10) can be written as:

$$\max\ \log(\frac{1}{2\Sigma^2}\exp(-\frac{|Y-\hat{Y}|}{\Sigma})) \tag{13}$$

Under pixel-wise scope, this can be expressed as:

$$\underset{\hat{y}_i,\sigma_i}{\operatorname{argmin}} \sum \frac{|y_i-\hat{y}_i|}{\sigma_i} - 2\log\frac{1}{\sigma_i}, \tag{14}$$

here, $\sigma_i \in \Sigma, y_i \in Y, \hat{y}_i \in \hat{Y}$. Since σ is provided by DCPlugin, Eq. (14) can be simplified as:

$$\underset{\hat{y}_i}{\operatorname{argmin}} \sum c_i|y_i-\hat{y}_i|, \tag{15}$$

$c_i = f(\sigma_i) \in C$, indicates the confidence map, optimize over $\hat{y}_i$ is equal to optimize over θ_d.

Therefore, the modelling of C should satisfy:

(1) $C = f(\Sigma)$, where f is a deterministic mapping function.
(2) $\Sigma = |\hat{Y}-Y|$, to align with the assumption of Laplace distribution.
(3) Physically speaking, the domain range of C should be $(0,1)$.

Based on the discussion above, to meet constraints, we propose using:

$$C = \alpha\exp(-\beta|\hat{Y}-Y|), \tag{16}$$

to model the confidence map. We set α and β to 1.

Channel-wise Style Transfer. Literature under DA [10,32] has widely discussed the importance of adaptation at the input level, which refers to style transformation from the source domain to the target domain, or vice versa. Mainstream pipelines [34,35] commonly utilize CycleGAN [47] for pixel-level adaptation. However, additional pre-training on CycleGAN can be overwhelming, and the quality is not assured. [48] discussed that style information within an image can be represented by its channel-wise mean–variance.

Specifically, suppose we have an image from source domain $img_s^{h,w,c} \in I_s$ and an image from target domain $img_t^{h,w,c} \in I_t$, we obtain translated source image $\hat{img}_s^{h,w,c} \in I_{s\to t}$ by:

$$\hat{img}_s^{h,w,c} = img\_norm_s^{h,w,c} \times \sigma_t^c + \mu_t^c. \tag{17}$$

σ_t^c and μ_t^c are the variance and mean over the target image's channel.

Loss Function. In this stage, both depth completion network and DCPlugin are trainable, and supervised by $\mathcal{L}_{vanilla}$ and $\mathcal{L}_I^{plugin}$ separately, where $\mathcal{L}_{vanilla}$ is the original loss used by depth completion network and $\mathcal{L}_I^{plugin}$ is represented as:

$$\mathcal{L}_I^{plugin} = \mathcal{L}_{bce}(\hat{C}_s, C_s) + w_{adv}\mathcal{L}_{adv}, \tag{18}$$

$$\mathcal{L}_{adv} = \sum_{i=1}^{N} -\log(\mathcal{D}(\mathbf{t}_i^I)). \tag{19}$$

C_s is calculated via Eq. (16), $\mathcal{D}$ is the discriminator, $\mathbf{t}_i^I$ is domain-invariant token on layer i and w_{adv} is used to balance the loss.

3.3. Stage II: Confidence-aware co-training

Objective. After obtaining DCPlugin and pre-trained depth completion network in Stage I, in this stage, we focus on two tasks: first, introducing *aleatoric* uncertainty to refine the predicted certainty map of the DCPlugin; second, utilizing the DCPlugin to adapt depth completion network.

Confidence Map Refinement. As discussed in Section 3.1, *epistemic* uncertainty is not enough, therefore, motivated by [41], we propose a confidence map refinement (CMR) scheme in Stage II by introducing uncertainty among the data.

Specially speaking, given input from target domain Y_t', I_t, we first feed it into MetaNet, which is structurally identical to depth completion network, and then DCPlugin, generating dense prediction $\hat{Y}_p$ and coarse confidence map $\hat{C}_t$ respectively. Afterwards, we use Eq. (17) to translate the image from the target domain to the style of the source domain, $I_{t\to s}$. For an input $Y_t', I_{t\to s}$, we perform a horizontal flip before forwarding it to the MetaNet, and another horizontal flip on the prediction as illustrated in Fig. 5.

Having obtained $\hat{Y}_p$ and $\hat{Y}_p^{flip}$, we model the compensation confidence map C_t^{flip} by Eq. (16). Our final confidence map is obtained by combining the coarse confidence map and the compensation confidence map with a weighted sum:

$$C_t^{fine} = w\cdot\hat{C}_t + (1-w)\cdot C_t^{flip}. \tag{20}$$

It is worth noting that the incorporation of $I_{t\to s}$ into this scheme aligns logically to reduce the domain gap within the common domain.

Adaptation through the Common Domain. After establishing the common domain through DCPlugin and CMR, we implement a teacher–student scheme to perform adaptation for depth completion network. Specifically, we treat MetaNet as the teacher and depth completion network as the student. Both MetaNet and depth completion network have identical structures and are initialized from Stage I. The parameters of depth completion network, denoted as θ_d^S, are learnable using Eq. (22), while the parameters of MetaNet, denoted as $\theta_d^{\mathcal{T}}$, are updated using the exponential moving average (EMA) [49] of θ_d^S:

$$\theta_d^{\mathcal{T}} = t\theta_d^{\mathcal{T}} + (1-t)\theta_d^S, \tag{21}$$

where t is the momentum coefficient ranging from 0 to 1, typically set close to 1 to ensure smooth parameter updates.

Loss Function. The depth completion network is updated using $\mathcal{L}_{II}^{dep}$:

$$\mathcal{L}_{II}^{dep} = w_1\mathcal{L}_{vanilla} + w_2\mathcal{L}_{l1}^t + w_3\mathcal{L}_{pseudo}^t, \tag{22}$$

$$\mathcal{L}_{l1}^t = \sum_{y_t'\in Y_t', \hat{y}_t\in\hat{Y}_t} |y_t'-\hat{y}_t|, \tag{23}$$

$$\mathcal{L}_{pseudo}^t = \sum_{c\in C_t^{fine}, \hat{y}_p\in\hat{Y}_p, \hat{y}_t\in\hat{Y}_t} c\cdot|\hat{y}_p-\hat{y}_t|. \tag{24}$$

$\mathcal{L}_{pseudo}^t$ is derived from Eq. (15) and the hyperparameter w_1, w_2, and w_3 are utilized to balance each loss.

4. Experiments

4.1. Datasets and evaluation metric

In this paper, we use three datasets. For the synthetic dataset, we utilize CARLA [15] and VKITTI [51,52]. For the real scene dataset, we employ KITTI [17]. All of our results are reported on the KITTI benchmark. As introduced in [38,53], we use Area Under the Sparsification Error (AUSE, the lower the better) and Area Under the Random Gain (AURG, the higher the better) to estimate the accuracy of the certainty map.

CARLA is collected in the simulator Carla [16], using the setup described in [14]. It consists of 18,022 images for training and 3,800 for validation.

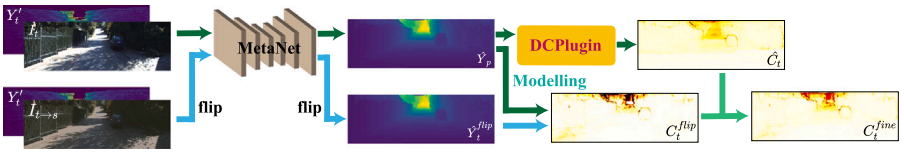


Fig. 5. Illustration of confidence map refinement: we address uncertainty from depth completion network and data using $\hat{C}_t$ and C_t^{flip}. The former is directly obtained from the trained DCPlugin, while the latter is derived from a post-processing step, specifically by flipping to estimate uncertainty.

Table 1
Comparison results on the KITTI benchmark: We present domain adaptation results for two adaptation scenarios, CARLA→KITTI and VKITTI→KITTI. Our method with FusionNet [3] achieves state-of-the-art performance in the domain adaptation category. The term 'Basic' refers to the approach of using only CycleGAN for domain adaptation (Fig. 2(a)), with the weights of CycleGAN obtained from [15] (see [50]).

Methods	Validation		Online Test				Category
	RMSE↓	MAE↓	RMSE↓	MAE↓	iRMSE↓	iMAE↓	
IP-Basic [50]	1350.93	305.35	1288.46	302.6	3.78	1.29	non DNN
DDP [4]	1310.03	347.17	1263.19	343.46	3.58	1.32	self-supervised
VOICED [26]	1239.06	305.06	1169.97	299.41	3.56	1.2	self-supervised
AdaFrame [27]	1189.43	298.89	1125.67	291.62	3.32	1.16	self-supervised
ScaffNet [29]	1182.81	286.35	1121.93	280.76	3.3	1.15	self-supervised
KBNet [25]	1128.01	262.01	1069.47	256.76	2.95	1.02	self-supervised
FusionNet [3]	**802**	**214**	772.87	215.05	2.19	0.93	supervised
BP-Net [22]	-	-	**684.90**	**194.69**	**1.82**	**0.84**	supervised
VKITTI → KITTI							
T2Net [35] + [5]	1789.06	416.77	-	-	-	-	DA
DCPlugin-U (Ours) + [5]	**1534.05**	**388.37**	-	-	-	-	DA
T2Net [35] + [3]	1916.47	407.30	-	-	-	-	DA
DCPlugin-U (Ours) + [3]	**1516.34**	**365.79**	-	-	-	-	DA
T2Net [35] + [6]	1774.29	371.40	-	-	-	-	DA
DCPlugin-U (Ours) + [6]	**1418.80**	**357.26**	-	-	-	-	DA
CARLA → KITTI							
SynthProj [15] + [5]	1208.62	339.63	-	-	-	-	DA
DCPlugin-U (Ours) + [5]	**1159.61**	**327.53**	-	-	-	-	DA
SynthProj [15] + [6]	1202.57	303.74	-	-	-	-	DA
DCPlugin-U (Ours) + [6]	**1168.47**	**303.41**	-	-	-	-	DA
SynthProj [15] + [3]	1150.27	281.94	1095.26	280.42	3.53	1.19	DA
Basic + [3]	1227.75	346.15	-	-	-	-	DA
DCPlugin-E (Ours) + [3]	1123.33	276.21	1077.10	274.75	3.33	1.16	DA
DCPlugin-U (Ours) + [3]	**1115.59**	**275.11**	**1069.88**	**272.55**	**3.45**	**1.18**	DA

VKITTI is a photo-realistic copy of the KITTI dataset generated on the Unity game engine. It contains 21,260 image-depth frames in 5 sequences, identical to KITTI but rendered with different weather conditions and camera viewpoints.

KITTI is a real-world collected dataset that consists of 86,898 image-depth frames for training, 7,000 for validation, and an official set of 1,000 frames for online testing, for which the ground truth is unreleased.

Evaluation Metric for Depth Completion. Following the KITTI benchmark, we choose RMSE (mm) as our main metric. For the validation set, we also report MAE. For the online test set, we additionally report iRMSE and MAE following prior works.

Evaluation Metric for Uncertainty. As introduced in [38,53], we use Area Under the Sparsification Error (AUSE, the lower the better) and Area Under the Random Gain (AURG, the higher the better) to estimate the accuracy of the certainty map. For each depth completion metric, AUSE and AURG are computed accordingly.

4.2. Training details

For every depth completion networks [3,5,6] we tested, we evaluated two adaptation scenarios: CARLA→KITTI and VKITTI→KITTI. We set the value of w in Eq. (20) to 0.5, t in Eq. (21) to 0.999, and w_{adv} in Eq. (18) to 0.001. We used the Adam optimizer with an initial learning rate of 1e-3 and batch size of 6. The experiments on FusionNet [3] and MSG-CHN [6] were conducted on a single RTX3090, while two RTX3090 GPUs were used for GuidedNet [5].

The learning rate decay strategies differ between Stage and depth completion network. In Stage I, only FusionNet utilizes a constant learning rate throughout all epochs, while the other models employ the CosineLR decay strategy. In Stage II, all models use a constant learning rate. The applied strategy remains consistent across the two adaptation scenarios.

For DCPlugin-U and E, we utilized the AdamW optimizer with an initial learning rate of 1e-4 and batch size of 6. The learning rate was decayed using CosineLR.

For the discriminator, we simply employed several linear layers with the activation function: $Linear_{128}^{192} - ReLU - Linear_{256}^{128} - ReLU - Linear_{1}^{256}$, since we only performed adversarial learning on DIT. It was optimized by Adam using the same initial learning rate and decay strategy as DCPlugin.

For Stage II, we defined one epoch as a complete iteration of the dataset from the source domain, as the volume of CARLA and VKITTI is much smaller than KITTI. For VKITTI→KITTI, we aligned all 5 scenarios in VKITTI to form an exactly one-to-one matching from VKITTI to KITTI, as implemented in T2Net, as well as in our reimplementation of T2Net, to reduce error. In both adaptation scenarios, we trained for 30 epochs for every depth completion networks. The values of w_1, w_2, and w_3 for FusionNet and GuideNet were set as $2, 2, 1$ and $1, 1, 0.5$ respectively, and remained consistent for CARLA and VKITTI. However, MSG-CHN had different values: $0.12, 2, 1$ for CARLA→KITTI and $0.2, 2, 1$ for VKITTI→KITTI.

For Stage I, the number of training epochs varied. We trained for 20 epochs for FusionNet on CARLA and VKITTI, 15 epochs for MSG-CHN and GuideNet on VKITTI, and 30 epochs and 20 epochs for MSG-CHN and GuideNet on CARLA.

For the reimplementation of T2Net, we set the batch size to 2 for FusionNet, 3 for MSG-CHN, and 4 for GuidedNet. The initial learning

rate was proportionally decayed according to the original practice. Similar to our pipeline, we used two GPUs for GuideNet and one for FusionNet and MSG-CHN. One key reason for the small batch size is the large memory consumption of T2Net, as multiple sub-networks need to be trained. From this perspective, our pipeline is effective and efficient.

4.3. Performance on benchmarks

We adapt our method with three depth completion models: GuideNet [5], MSG-CHN [6], and FusionNet [3].

We perform domain adaptation in two scenarios: CARLA→KITTI and VKITTI→KITTI. Since there is little work on the UDA of depth completion, we also list some self-supervised works for reference, as shown in Table 1.

CARLA→KITTI. For each depth completion model, our method outperforms the previous DA method [15] and almost all self-supervised methods except KBNet. For FusionNet, our method achieves state-of-the-art performance under the DA method and performs comparably to KBNet on the KITTI online benchmark, surpassing SynthProj by a significant margin.

VKITTI→KITTI. Projection artifacts cannot be generated in this adaptation scenario because the sensor location is fixed in VKITTI but editable in CARLA. As a result, we cannot compare our results with SynthProj, and performance is generally lower than CARLA→KITTI. Alternatively, we implemented the well-known domain adaptation method for depth estimation, T2Net [35], for comparison. From Table 1, we can see that our method significantly outperforms T2Net.

Comparison with self-supervised Methods. Our methods outperform all self-supervised methods except slightly lower than KBNet [25], we hereby present analysis below:

- Our adaptation scenario performs from a low-volume dataset (20k on CARLA) to a high-volume dataset (90k on KITTI). It is natural to experience lower performance, and it is an open challenge in this realm.
- Self-supervised methods are not counterparts to domain adaptation methods. Theoretically, better performance can be achieved by incorporating the self-supervised photometric loss [2] into co-training on the target domain, as confirmed by SynthProj. However, we omit this approach here, as tuning the hyperparameter is time-consuming and not relevant to the contributions of this paper.

Comparison with supervised methods. Although our method demonstrates promising performance, it remains significantly behind SOTA supervised approaches on the KITTI online leaderboard, where the current SOTA supervised method [22] achieves an RMSE of approximately 685. This performance disparity can primarily be attributed to the limited size of our training dataset; the source domain utilized in our approach contains only 18,022 samples, compared to the 86,898 samples available in the KITTI dataset.

4.4. Ablation studies

4.4.1. Overall

All the ablation studies were conducted using FusionNet on the CARLA→KITTI adaptation scenario. The evaluation results are reported on the KITTI validation benchmark.

First, we present comprehensive ablation studies of our framework in Table 2, which contains contributions from each of the components mentioned in this paper, including domain-invariant token (**DIT.**), confidence map refinement (**CMR.**) and CWST. We conducted our experiments using DCPlugin-U and DCPlugin-E to provide more generalized results. As shown in Table 2, DCPlugin-U is better than DCPlugin-E in terms of overall performance. Domain-invariant tokens can slightly improve RMSE performance on StageII when combined with CMR and CWST in both types of DCPlugin. The rest of the components will be investigated in subsequent sections, and the presentation of the overall framework provides a basis for further experimentation and analysis. Specifically, the experimental setups for Tables 5 and 6 can be found in the corresponding serial numbers displayed in Table 4, while the specific experimental setups represented by the serial numbers can be indexed to a specific row in Table 2.

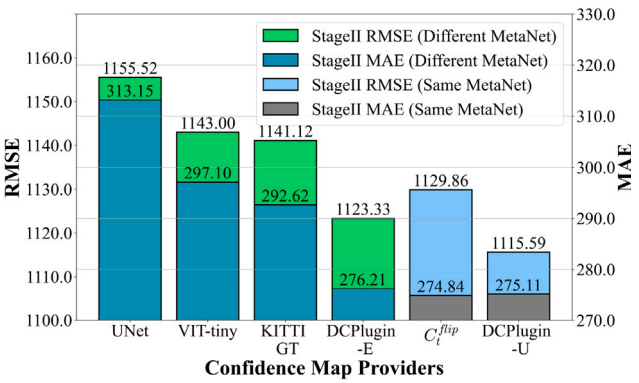


Fig. 6. Ablation studies on the source of the confidence map. 'C_t^{flip}' equals to setting w in Eq. (20) to 0.

4.4.2. Uncertainty analysis

To verify the effect of introducing uncertainty, we performed ablation experiments on CWST and the basic teacher–student paradigm, using SythnProj [15] (previous SOTA) as a baseline. As Table 3 shows, the introduction of uncertainty (5th, 6th row) has significantly improved the performance not only for the RMSE but also for the MAE when comparing to style-trans and student–teacher, individually (2nd, 3rd row) or in combination (4th row).

Furthermore, when incorporating Student–Teacher (3rd, 4th row), despite improvement on RMSE, MAE still poses worse than previous SOTA, which means the model can predict accurate in long-range distance while overlooking the close region. This imbalance roots from the lack of uncertainty since close region objects cannot be treated equally due to the observation demonstrated on Fig. 1 (prediction on close range objects' edge is worse but generally accurate), therefore emphasizing the importance of introducing uncertainty

Another bonus for involving uncertainty is the robustness of source-train weights. The performance of the final results for the basic Student–Teacher scheme can be affected by the initiation weights of the teacher model. 5th row from Table 3 indicates that even with a much worse teacher, our method still exhibits superior performance compared with the previous SOTA.

4.4.3. Source of confidence map

To assess the effectiveness of our DCPlugin, we investigate different providers of the confidence map, as illustrated in Fig. 6. We obtain the confidence map from six different providers: U-Net [20], VIT-tiny [43], KITTI ground truth, C_t^{fine} from DCPlugin-E, C_t^{flip} from Eq. (20), and C_t^{fine} from DCPlugin-U. The experiment reveals two important observations: 1. The introduction of uncertainty contributes to improved performance. 2. Enhanced construction of uncertainty results in even better performance.

It is worth noting that even when using KITTI ground truth to generate the confidence map, the performance is still not promising. This is because the obtained confidence map is semi-dense (near 30% density on KITTI).

Table 2
Comprehensive ablation studies of our framework. T., DIT. under *DCPlugin* stands for Type, Domain-invariant Token respectively, CMR. stands for Confidence Map Refinement. CWST used on with mark represents it is validated during Stage I and Stage II.

ID	*DCPlugin*		CMR.	CWST	Depth Metric			
					StageI		*StageII*	
	T.	DIT.			RMSE↓	MAE↓	RMSE↓	MAE↓
1	E	✓	✓		1279.24	339.97	1135.95	280.23
2				✓	1216.63	317.69	1146.38	282.49
3			✓	✓			1127.11	283.59
4		✓		✓	1233.92	316.4	1134.69	277.47
5		✓	✓	✓			1123.33	276.21
6	U	✓	✓		1261.94	318.26	1135.06	284.85
7				✓	1219.74	305.43	1130.35	276.28
8			✓	✓			1122.39	272.91
9		✓		✓	1206.27	312.48	1133.09	282.66
10		✓	✓	✓			1115.59	275.11

Table 3
Ablation studies over our method against CWST (Style-trans) and basic Student–Teacher paradigm.

Arch: FusionNet	Source-train weights		RMSE ↓	MAE ↓
	RMSE ↓	MAE ↓		
SythnProj (previous SOTA)	–	–	1150.27	281.94
Style-trans	–	–	1156.41 (+6.14)	306.68 (+24.74)
Student–Teacher	1285.41	329.69	1149.25 (−1.02)	307.33 (+25.39)
Student–Teacher + Style-trans	1206.27	312.48	1132.21 (−18.06)	292.25 (+10.31)
Ours	1285.41	329.69	1129.89 (−20.38)	278.73 (−3.21)
+ Style-trans	1206.27	312.48	1115.59 (−34.48)	275.11 (−6.83)

Table 4
Setups for Table 5 and Table 6. The setup for each experiment can be found on Table 2 accordingly.

Exp. reported on Table 5	Setting I	Setting II	Setting III	Setting IV
ID on Table 2	2	4	7	9
Exp. reported on Table 6	2nd Row	3th Row	4th Row	5th Row
ID on Table 2	1	3	6	8

4.4.4. Confidence map from depth completion network

Works like [3,44,54] do provide uncertainty alone with their prediction, however, we do not consider that in our work for the following reasons:

- Physically incompatible. In [3], the uncertainty is from different modalities rather than the final dense prediction, which means further tuning needs to be done to incorporate it into our framework.
- Low versatility. Uncertainty from some uncertainty-based pipelines like [44,54] can be directly used. However, this kind of modification or model-level adaptation is contrary to our original vision, which is versatility, as we cannot force every DC model designed via uncertainty or a particular paradigm.

4.4.5. Effectiveness of confidence map refinement

To evaluate the effectiveness of CMR, we present four different settings. Details of each setting can be cross-referenced using Tables 2 and 4. Numerical results are provided in Table 5. In each setting, after adding CMR, both the depth metric and the confidence metric show improvement.

4.4.6. Robustness to image transformation

Most domain adaptation works in the field of depth estimation [32,35] or depth completion [15] heavily rely on image transformation approaches to minimize the gap between the source domain and the target domain. To evaluate the robustness of our framework, we conducted comprehensive ablation studies as shown in Table 6. We set only use CWST as a baseline on the first row, comparing 1st, 2nd row and 1st, 4th row, it can be observed that even without applying style transfer during Stage I and Stage II, our framework performs better than the transferred method. Furthermore, when we perform image transformation on both stages, cooperating with $\mathcal{L}^{t}_{pseudo}$, we observe further improvement (3rd row and 5th row). Besides, we also investigate the robustness of CMR when CWST is not presented, as shown in Table 7. It can be seen that the CWST on CMR can be replaced by data augmentation techniques, e.g. colour jitters. Since the presence of CWST on CMR contributes little fluctuation compared to the full pipeline, we conclude that the performance gain roots from the introduction of uncertainty rather than CWST.

4.4.7. Long-term robustness on stage II

To showcase the efficiency of our training framework, we present per-epoch evaluation results in the adaptation scenario CARLA→KITTI for depth completion network GuidedNet and MSG-CHN. These results are compared with those obtained using SynthProj, as depicted in Fig. 7. It is evident that our training framework demonstrates a relatively robust resistance to overfitting compared to SynthProj. As the number of iterations increases, the evaluation results of our method exhibit a positive trend, whereas the performance of SynthProj deteriorates.

4.5. Visualization results

Finally, to empirically validate the capabilities of the framework, we visualize the depth completion results obtained by DCPlugin and other methods in Fig. 8. Specifically, in Fig. 8(a), we compare the Visualization results of FusionNet [3] with DCPlugin and SynthProj [15] for CARLA to KITTI. In Fig. 8(b), we show the difference in prediction and error between MSG-CHN [6] with DCPlugin and T2Net [35] in VKITTI to KITTI. Colour marks demonstrate the main difference.

In Fig. 8(a) 1st row, our method achieved more precise outlines of the barrier gate arm compared to SynthProj. Additionally, in Fig. 8(a) 2nd, our method treats the depth of glass as a solid entity, similar to the vehicle, rather than focusing on its interior.

Table 5
Effectiveness of the confidence map refinement (CMR). We conducted experiments in four different settings. In each setting, improvements can be observed after adding CMR.

Arch: FusionNet	Depth metric	Confidence metric			
		RMSE		MAE	
	RMSE↓	**AUSE↓**	**AURG↑**	**AUSE↓**	**AURG↑**
Setting I	1146.38	0.1251	1.0090	0.0500	0.2114
+ CMR	1127.11 (-19.27)	0.1114 ↓	1.0226 ↑	0.0479 ↓	0.2136 ↑
Setting II	1134.69	0.1194	1.0314	0.0465	0.2118
+ CMR	1123.33 (-11.36)	0.1053 ↓	1.0454 ↑	0.0499 ↓	0.2134 ↑
Setting III	1130.35	0.4030	0.7386	0.1236	0.1288
+ CMR	1122.39 (-7.96)	0.2304 ↓	0.9127 ↑	0.0903 ↓	0.1621 ↑
Setting IV	1133.09	0.2117	0.9127	0.0633	0.1927
+ CMR	1115.59 (-17.5)	0.1551 ↓	0.9694 ↑	0.0559 ↓	0.2002 ↑

Table 6
Robustness of confidence-aware co-training. 'CWST' stands for channel-wise style transfer. The experiment demonstrates that the performance gain is rooted in the introduction of uncertainty on both types (2nd, 4th row).

Type	CWST	$\mathcal{L}^{t}_{pseudo}$	Source-train weights		RMSE	MAE
			RMSE	MAE		
–	✓		1216.63	317.69	1156.41	306.68
E		✓	1279.24	339.97	1135.95 (−20.46)	280.23 (−26.45)
	✓	✓	1216.63	317.69	1127.11 (−29.30)	283.59 (−23.09)
U		✓	1261.94	305.43	1135.06 (−21.35)	284.85 (−21.83)
	✓	✓	1219.74	305.43	1122.39 (−34.02)	272.91 (−33.77)

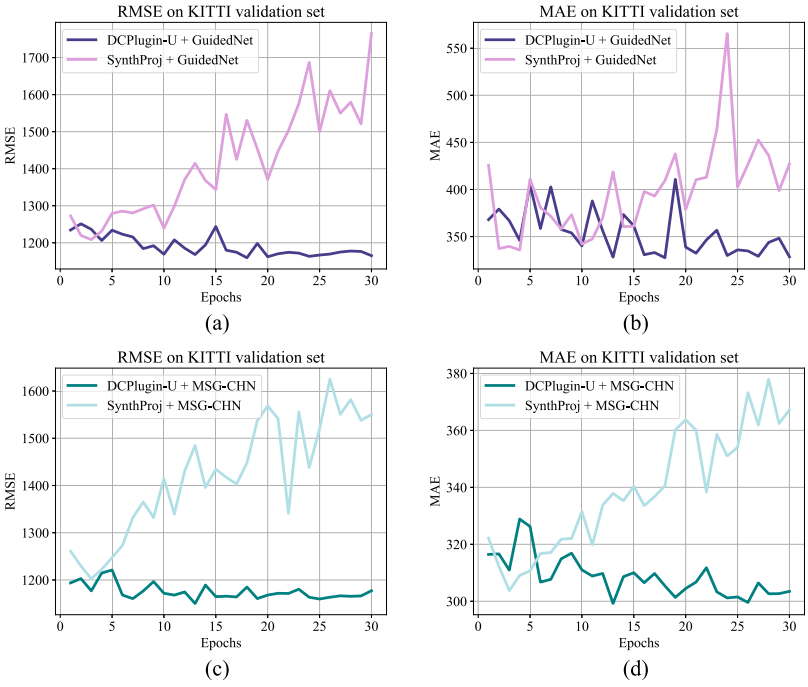


Fig. 7. Per-epoch evaluation results on GuidedNet and MSG-CHN. We compare our method with [15], the adaption scenario is CARLA→KITTI. The metric is reported under RMSE and MAE, the lower the better, best view in zoom.

Table 7
Robust analysis of CWST against CMR. When CWST on CMR is replaced or solely activated, the performance is still promising, posing that the effectiveness comes from the uncertainty.

Arch: FusionNet	RMSE	MAE
Proposed CMR	1129.89	278.73
replace CWST with data augmentation	1132.46	280.87
only CWST on CMR	1132.99	283.29

In Fig. 8(b), the error maps in the first three rows demonstrate that the red errors within the highlighted box are significantly fewer using our method. In the depth completion map of the 4th row, the central road is devoid of cars, yet T2Net estimates a blurry car outline, and its prediction for the sky blends with the surrounding trees. Conversely, our method distinctly differentiates the sky from the trees, with the sky's depth colour being very light, indicating a considerable distance. In the depth completion map of the 5th row, our method reveals more refined depth details in the distant cyclist and better-fitting tree outlines, whereas T2Net entirely lacks depth details for the distant cyclist, rendering a blurry and imprecise representation of tree outlines.

These observations highlight that, due to the introduction of uncertainty, our framework provides better completion results for both object edges and distant objects, with relatively smaller prediction errors. In contrast, previous methods could not balance these aspects effectively. Moreover, we provide Visualization results of the confidence

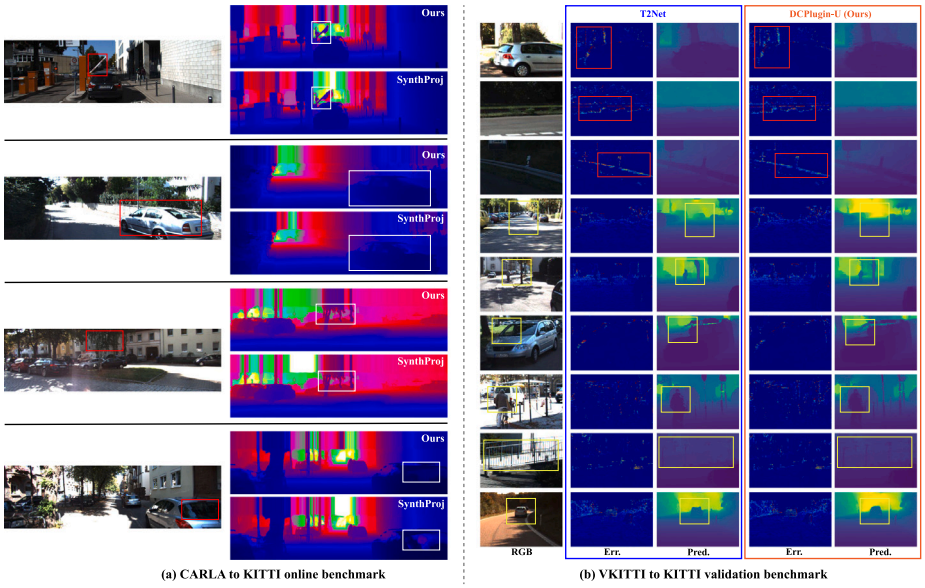


Fig. 8. Comparison Visualization results. (a) Visualization results on CARLA→KITTI of FusionNet, tested on KITTI online benchmark. We highlight the major difference between SynthProj and Ours. (b) Visualization results on VKITTI→KITTI of MSG-CHN, tested on KITTI validation set. The red and yellow marks demonstrate the difference between T2Net and ours. Best view in zoom. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

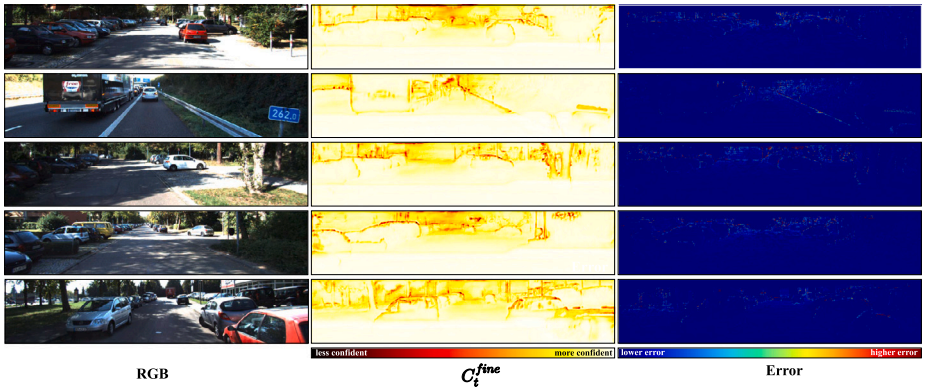


Fig. 9. Visualization of confidence map on CARLA→KITTI, tested on KITTI validation set using FusionNet. Pixels that lack ground truth in the error map are set to zero.

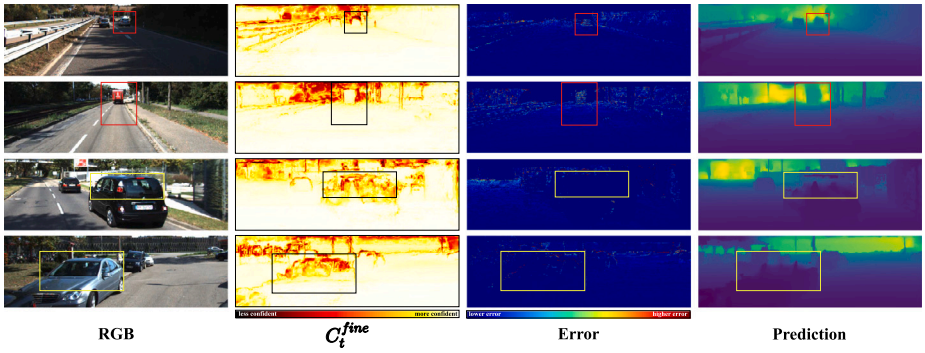


Fig. 10. Visualization of failure cases on CARLA→KITTI, tested on KITTI validation set using FusionNet. Pixels that lack ground truth in the error map are set to zero. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

map in Fig. 9, which clearly shows that the predicted confidence map accurately describes the property of spatial-error consistency.

The performance of our method is influenced by the imperfections in the predicted confidence map, as illustrated in Fig. 10. Specifically, the top two rows demonstrate that overconfidence in regions such as distant objects (e.g., the centre of the vehicle) can lead to inaccuracies, highlighted by the red ovals. Conversely, the bottom two rows reveal that areas consistently exhibiting low confidence (highlighted by yellow ovals) may result in incomplete predictions.

5. Conclusion

In this paper, we propose a novel training framework for unsupervised domain adaptation in depth completion. The framework leverages

the spatial-error consistency pattern by introducing the Depth Completion Plugin (DCPlugin) sub-module. Experimental results on multiple benchmarks demonstrate the effectiveness of the approach, outperforming other methods and achieving state-of-the-art performance. However, there are still limitations that need to be addressed for future research:

- The performance still is far behind supervised approaches due to the limited size of training data, future research could investigate scaling up both model parameters and training data volume to more comprehensively evaluate the potential of our method.
- Our method is affected by the issue of overconfidence of the confidence map, future work could explore mitigating this effect by incorporating model-specific uncertainty through established uncertainty estimation techniques [39,40].
- Light reflections and transparent materials may have low confidence on the confidence map during Stage II, resulting in incomplete predictions due to insufficient pseudo-labelling. A potential solution is to apply dense pseudo-supervision in these regions, possibly by utilizing depth estimation.

CRediT authorship contribution statement

Lingyu Xiao: Writing – review & editing, Writing – original draft, Software, Formal analysis. **Jinhui Wu:** Writing – review & editing. **Junjie Hu:** Writing – review & editing. **Ziyu Li:** Writing – review & editing. **Wankou Yang:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Nos. 62276061 and 62436002. This work is also supported by Research Fund for Advanced Ocean Institute of Southeast University (Major Program MP202404).

Data availability

Data will be made available on request.

References

[1] Y. Zhang, X. Guo, M. Poggi, Z. Zhu, G. Huang, S. Mattoccia, Completionformer: Depth completion with convolutions and vision transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 18527–18536.

[2] F. Ma, G.V. Cavalheiro, 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.

[3] W. Van Gansbeke, D. Neven, B. De Brabandere, 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.

[4] Y. Yang, A. Wong, S. Soatto, Dense depth posterior (ddp) from single image and sparse range, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3353–3362.

[5] J. Tang, F.-P. Tian, W. Feng, J. Li, P. Tan, Learning guided convolutional network for depth completion, IEEE Trans. Image Process. 30 (2020) 1116–1129.

[6] A. Li, Z. Yuan, Y. Ling, W. Chi, C. Zhang, et al., A multi-scale guided cascade hourglass network for depth completion, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 32–40.

[7] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, T. Darrell, Cycada: Cycle-consistent adversarial domain adaptation, in: International Conference on Machine Learning, Pmlr, 2018, pp. 1989–1998.

[8] T.-H. Vu, H. Jain, M. Bucher, M. Cord, P. Pérez, Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2517–2526.

[9] M. Chen, H. Xue, D. Cai, Domain adaptation for semantic segmentation with maximum squares loss, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2090–2099.

[10] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, M. Chandraker, Learning to adapt structured output space for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7472–7481.

[11] J. Deng, W. Li, Y. Chen, L. Duan, Unbiased mean teacher for cross-domain object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4091–4101.

[12] S. Li, M. Ye, X. Zhu, L. Zhou, L. Xiong, Source-free object detection by learning to overlook domain style, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8014–8023.

[13] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7167–7176.

[14] M. Long, Z. Cao, J. Wang, M.I. Jordan, Conditional adversarial domain adaptation, Adv. Neural Inf. Process. Syst. 31 (2018).

[15] A. Lopez-Rodriguez, B. Busam, K. Mikolajczyk, Project to adapt: Domain adaptation for depth completion from noisy and sparse sensor data, Int. J. Comput. Vis. 131 (3) (2023) 796–812.

[16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: An open urban driving simulator, in: Conference on Robot Learning, PMLR, 2017, pp. 1–16.

[17] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, A. Geiger, Sparsity invariant cnns, in: 2017 International Conference on 3D Vision, 3DV, IEEE, 2017, pp. 11–20.

[18] J. Hu, C. Bao, M. Ozay, C. Fan, Q. Gao, H. Liu, T.L. Lam, Deep depth completion from extremely sparse data: A survey, IEEE Trans. Pattern Anal. Mach. Intell. (2022).

[19] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[20] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.

[21] S. Liu, S. De Mello, J. Gu, G. Zhong, M.-H. Yang, J. Kautz, Learning affinity via spatial propagation networks, Adv. Neural Inf. Process. Syst. 30 (2017).

[22] J. Tang, F.-P. Tian, B. An, J. Li, P. Tan, Bilateral propagation network for depth completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 9763–9772.

[23] W. Wang, S. Chen, Y. Xiang, J. Sun, H. Li, Z. Wang, F. Sun, Z. Ding, B. Li, Sparsely-labeled source assisted domain adaptation, Pattern Recognit. 112 (2021) 107803.

[24] Z. Yan, Y. Lin, K. Wang, Y. Zheng, Y. Wang, Z. Zhang, J. Li, J. Yang, Tri-perspective view decomposition for geometry-aware depth completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 4874–4884.

[25] A. Wong, S. Soatto, Unsupervised depth completion with calibrated backprojection layers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12747–12756.

[26] A. Wong, X. Fei, S. Tsuei, S. Soatto, Unsupervised depth completion from visual inertial odometry, IEEE Robot. Autom. Lett. 5 (2) (2020) 1899–1906.

[27] A. Wong, X. Fei, B.-W. Hong, S. Soatto, An adaptive framework for learning unsupervised depth completion, IEEE Robot. Autom. Lett. 6 (2) (2021) 3120–3127.

[28] Z. Yan, K. Wang, X. Li, Z. Zhang, J. Li, J. Yang, Desnet: Decomposed scale-consistent network for unsupervised depth completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 3, 2023, pp. 3109–3117.

[29] A. Wong, S. Cicek, S. Soatto, Learning topology from synthetic data for unsupervised depth completion, IEEE Robot. Autom. Lett. 6 (2) (2021) 1495–1502.

[30] S. Nedevschi, et al., Online cross-calibration of camera and lidar, in: 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing, ICCP, IEEE, 2017, pp. 295–301.

[31] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (59) (2016) 1–35.

[32] A. Atapour-Abarghouei, 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.

[33] J.N. Kundu, P.K. Uppala, A. Pahuja, R.V. 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.

[34] S. Zhao, H. Fu, M. Gong, D. Tao, Geometry-aware symmetric domain adaptation for monocular depth estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9788–9798.
[35] C. Zheng, T.-J. Cham, 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.
[36] Y.-T. Yen, C.-N. Lu, W.-C. Chiu, Y.-H. Tsai, 3D-PL: Domain adaptive depth estimation with 3D-aware pseudo-labeling, in: European Conference on Computer Vision, Springer, 2022, pp. 710–728.
[37] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision? Adv. Neural Inf. Process. Syst. 30 (2017).
[38] M. Poggi, F. Aleotti, F. Tosi, S. Mattoccia, On the uncertainty of self-supervised monocular depth estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3227–3237.
[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[40] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, Adv. Neural Inf. Process. Syst. 30 (2017).
[41] C. Godard, O. Mac Aodha, 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.
[42] J. Hornauer, V. Belagiannis, Gradient-based uncertainty for monocular depth estimation, in: European Conference on Computer Vision, Springer, 2022, pp. 613–630.
[43] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021, URL https://openreview.net/forum?id=YicbFdNTTy.
[44] Y. Zhu, W. Dong, L. Li, J. Wu, X. Li, G. Shi, Robust depth completion with uncertainty-driven loss functions, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 3, 2022, pp. 3626–3634.
[45] M. Figueiredo, Adaptive sparseness using Jeffreys prior, Adv. Neural Inf. Process. Syst. 14 (2001).
[46] S. Shao, Z. Pei, W. Chen, R. Li, Z. Liu, Z. Li, Urcdc-depth: Uncertainty rectified cross-distillation with cutflip for monocular depth estimation, IEEE Trans. Multimed. (2023).
[47] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2223–2232.
[48] Z. Zhong, Y. Zhao, G.H. Lee, N. Sebe, Adversarial style augmentation for domain generalized urban-scene segmentation, Adv. Neural Inf. Process. Syst. 35 (2022) 338–350.
[49] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, Adv. Neural Inf. Process. Syst. 30 (2017).
[50] J. Ku, A. Harakeh, S.L. Waslander, In defense of classical image processing: Fast depth completion on the cpu, in: 2018 15th Conference on Computer and Robot Vision, CRV, IEEE, 2018, pp. 16–22.
[51] Y. Cabon, N. Murray, M. Humenberger, Virtual KITTI 2, 2020, arXiv:2001.10773.
[52] A. Gaidon, Q. Wang, Y. Cabon, E. Vig, Virtual worlds as proxy for multi-object tracking analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4340–4349.
[53] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, T. Brox, Uncertainty estimates and multi-hypotheses networks for optical flow, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 652–667.
[54] A. Eldesokey, M. Felsberg, K. Holmquist, M. Persson, Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12014–12023.

Lingyu Xiao is a Master student at the School of Automation, Southeast University. He received the Bachelor's degree in the School of Electronic and Control Engineering from Chang'an University in 2022. His research interests include computer vision and pattern recognition.

Jinhui Wu is a Master student at the School of Automation, Southeast University. He received the B.S. degree from the School of Automation, Northwestern Polytechnical University, Xi'an, China, in 2023. His current research interests include deep learning and computer vision.

Junjie Hu (Member, IEEE) received the M.S. and Ph.D. degrees from the Graduate School of Information Science, Tohoku University, Sendai, Japan, in 2017 and 2020, respectively. He is currently a research scientist with the Shenzhen Institute of Artificial Intelligence and Robotics for Society. His research interests include machine learning, computer vision, and robotics. He has published more than 30 research papers in top-tier international journals and conference proceedings in AI and robotics, such as TPAMI, TRO, ICCV, IJCAI, RAL, ICRA, and IROS. He serves as a reviewer of IJCV, TIP, CVPR, ECCV, ICRA, IROS, RAL, IEEE Transactions on Mechatronics, Neurocomputing, etc.

Ziyu Li is a Ph.D. student at the School of Automation, Southeast University. He received the Bachelor's degree in the College of Astronautics from Nanjing University of Aeronautics and Astronautics in 2018 and finished the Master courses in the School of Automation, Southeast University in 2020 spring. His research interests include computer vision, pattern recognition, especially the scene understanding in autonomous driving scenarios and 3D object detection with point clouds.

Wankou Yang received the B.S., M.S. and Ph.D. degrees in the School of Computer Science and Technology, Nanjing University of Science and Technology (NUST), China, respectively in 2002, 2004, and 2009. From July 2009 to Aug. 2011, he worked as a Postdoctoral Fellow in the School of Automation, Southeast University, China. He is the professor in School of Automation, Southeast University. His research interests include pattern recognition, computer vision and machine learning.