3D-PL: Domain Adaptive Depth Estimation with 3D-aware Pseudo-Labeling

Yu-Ting Yen[1,2], Chia-Ni Lu[1], Wei-Chen Chiu[1], Yi-Hsuan Tsai[2]

[1]National Chiao Tung University, Taiwan [2]Phiar Technologies

Abstract. For monocular depth estimation, acquiring ground truths for real data is not easy, and thus domain adaptation methods are commonly adopted using the supervised synthetic data. However, this may still incur a large domain gap due to the lack of supervision from the real data. In this paper, we develop a domain adaptation framework via generating reliable pseudo ground truths of depth from real data to provide direct supervisions. Specifically, we propose two mechanisms for pseudo-labeling: 1) 2D-based pseudo-labels via measuring the consistency of depth predictions when images are with the same content but different styles; 2) 3D-aware pseudo-labels via a point cloud completion network that learns to complete the depth values in the 3D space, thus providing more structural information in a scene to refine and generate more reliable pseudo-labels. In experiments, we show that our pseudo-labeling methods improve depth estimation in various settings, including the usage of stereo pairs during training. Furthermore, the proposed method performs favorably against several state-of-the-art unsupervised domain adaptation approaches in real-world datasets. Our code and models are available at https://github.com/ccc870206/3D-PL.

Keywords: domain adaptation, monocular depth estimation, pseudo-labeling

1 Introduction

Monocular depth estimation is an ill-posed problem that aims to estimate depth from a single image. Numerous supervised deep learning methods [3,9,12,23,30,52] have made great progress in recent years. However, they need a large amount of data with ground truth depth, while acquiring such depth labels is highly expensive and time-consuming because it requires depth sensors such as LiDAR [15] or Kinect [55]. Therefore, several unsupervised methods [14,16,17,33,46,54] have been proposed, where these approaches estimate disparity from videos or binocular stereo images without any ground truth depth. Unfortunately, since there is no strong supervision provided, unsupervised methods may not do well under situations such as occlusion or blurring in object motion. To solve this problem, recent works use synthetic datasets since the synthetic image-depth pairs are easier to obtain and have more accurate dense depth information than real-world depth maps. However, there still exists domain shift between synthetic

(a) Overview of our proposed method for domain adaptation.

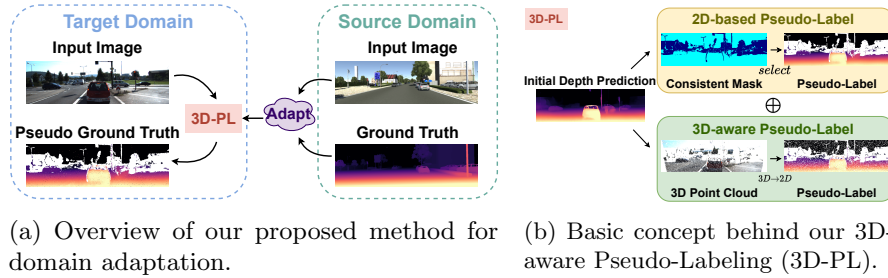(b) Basic concept behind our 3D-aware Pseudo-Labeling (3D-PL).

Fig. 1: **(a)** We propose a 3D-aware pseudo-labeling (3D-PL) technique to facilitate source-to-target domain adaptation for monocular depth estimation via pseudo-labeling on the target domain. **(b)** Our 3D-PL technique consists of 2D-based and 3D-aware pseudo-labels, where the former selects the pixels with highly-confident depth prediction (colorized by light blue in the consistent mask), while the latter performs 3D point cloud completion that provides refined pseudo-labels projected from 3D.

and real datasets, and thus many works use domain adaptation [6, 25, 31, 56, 58] to overcome this issue.

In the scenario of domain adaptation, two major techniques are commonly adopted to reduce the domain gap for depth estimation: 1) using adversarial loss [6, 25, 58] for feature-level distribution alignment, or 2) leveraging style transfer between synthetic/real data to generate real-like images as pixel-level adaptation [56, 58]. On the other hand, self-learning via pseudo-labeling the target real data is another powerful technique for domain adaptation [29, 47, 60], yet less explored in the depth estimation task. One reason is that, unlike tasks such as semantic segmentation that has the probabilistic output for classification to produce pseudo-labels, depth estimation is a regression task which requires specific designs for pseudo-label generation. In this paper, we propose novel pseudo-labeling methods in depth estimation for domain adaptation (see Fig. 1a).

To this end, we propose two mechanisms, 2D-based and 3D-aware methods, for generating pseudo depth labels (see Fig. 1b). For the 2D type, we consider the consistency of depth predictions when the model sees two images with the same content but different styles, i.e., the depth prediction can be more reliable for pixels with higher consistency. Specifically, we apply style transfer [20] to the target real image and generate its synthetic-stylized version, and then find their highly-consistent areas in depth predictions as pseudo-labels. However, this design may not be sufficient as it produces pseudo-labels only in certain confident pixels but ignore many other areas. Also, it does not take the fact that depth prediction is a 3D task into account.

To leverage the confident information obtained in our 2D-based pseudo-labeling process, we further propose to find the neighboring relationships in the 3D space via point cloud completion, so that our model is able to even select the pseudo-labels in areas that are not that confident, thus being complementary to

2D-based pseudo-labels. Specifically, we first project 2D pseudo-labels to point clouds in the 3D space, and then utilize a 3D completion model to generate neighboring point clouds. Due to the help of more confident and accurate 2D pseudo-labels, it also facilitates 3D completion to synthesize better point clouds. Next, we project the completed point clouds back to depth values in the 2D image plane as our 3D-aware pseudo-labels. Since the 3D completion model learns the whole structural information in 3D space, it can produce reliable depth values that correct the original 2D pseudo-labels or expand extra pseudo-labels outside of the 2D ones. We also note that, although pseudo-labeling for depth has been considered in the prior work [31], different from this work that needs a pre-trained panoptic segmentation model and can only generate pseudo-labels for object instances, our method does not have this limitation as we use the point cloud completion model trained on the source domain to infer reliable 3D-aware pseudo-labels on the target image.

We conduct extensive experiments by using the virtual KITTI dataset [13] as the source domain and the KITTI dataset [15] as the real target domain. We show that both of our 2D-based and 3D-aware pseudo-labeling strategies are complementary to each other and improve the depth estimation performance. In addition, following the stereo setting in GASDA [56] where the stereo pairs are provided during training, our method can further improve the baselines and perform favorably against state-of-the-art approaches. Moreover, we directly evaluate our model on other unseen datasets, Make3D [43], and show good generalization ability against existing methods. Here are our main contributions:

– We propose a framework for domain adaptive monocular depth estimation via pseudo-labeling, consisting of 2D-based and 3D-aware strategies that are complementary to each other.
– We utilize the 2D consistency of depth predictions to obtain initial pseudo-labels, and then propose a 3D-aware method that adopts point cloud completion in the structural 3D space to refine and expand pseudo-labels.
– We show that both of our 2D-based and 3D-aware methods have advantages against existing methods on several datasets, and when having stereo pairs during training, the performance can be further improved.

2 Related Work

Monocular Depth Estimation. Monocular depth estimation is to understand 3D depth information from a single 2D image. With the recent renaissance of deep learning techniques, supervised learning methods [3, 9, 12, 23, 30, 52] have been proposed. Eigen *et al.* [9] first use a two-scale CNN-based network to directly regress on the depth, while Liu *et al.* [30] utilize continuous CRF to improve depth estimation. Furthermore, some methods propose different designs to extend the CNN-based network, such as changing the regression loss to classification [3,12], adding geometric constraints [52], and predicting with semantic segmentation [23, 48].

Despite having promising results, the cost of collecting image-depth pairs for supervised learning is expensive. Thus, several unsupervised [14, 16, 17, 33, 46, 54] or semi-supervised [1, 18, 22, 26] methods have been proposed to estimate disparity from the stereo pairs or videos. Garg *et al.* [14] warp the right image to reconstruct its corresponding left one (in a stereo pair) through the depth-aware geometry constraints, and take photometric error as the reconstruction penalty. Godard *et al.* [16] predict the left and right disparity separately, and enforce the left-right consistency to enhance the quality of predicted results. There are several follow-up methods to further improve the performance through semi-supervised manner [1, 26] and video self-supervision [17, 33].

Domain Adaptation for Depth Estimation. Another way to tackle the difficulty of data collection for depth estimation is to leverage the domain adaptation techniques [6, 25, 31, 38, 56, 58], where the synthetic data can provide full supervisions as the source domain and the real-world unlabeled data is the target domain. Since depth estimation is a regression task, existing methods usually rely on style transfer/image translation for pixel-level adaptation [2], adversarial learning for feature-level adaptation [25], or their combinations [56, 58]. For instance, Atapour *et al.* [2] transform the style of testing data from real to synthetic, and use it as the input to their depth prediction model that is only trained on the synthetic data. AdaDepth [25] aligns the distribution between the source and target domain at the latent feature space and the prediction level. T^2net [58] further combines these two techniques, where they adopt both the synthetic-to-real translation network and the task network with feature alignment. They show that, training on the real stylized images brings promising improvement, but aligning features is not effective in the outdoor dataset.

Other follow-up methods [6, 56] take the bidirectional translation (real-to-synthetic and synthetic-to-real) and use the depth consistency loss on the prediction between the real and real-to-synthetic images. Moreover, some methods employ additional information to give constraints on the real image. GASDA [56] utilizes stereo pairs and encourages the geometry consistency to align stereo images. With a similar setting and geometry constraint to GASDA, SharinGAN [38] maps both synthetic and real images to a shared image domain for depth estimation. Moreover, DESC [31] adopts instance segmentation to apply pseudo-labeling using instance height and semantic segmentation to encourage the prediction consistency between two domains. Compared to these prior works, our proposed method provides direct supervisions on the real data in a simple and efficient pseudo-labeling way without any extra information.

Pseudo-Labeling for Depth Estimation. In general, pseudo-labeling explores the knowledge learned from labeled data to infer pseudo ground truths for unlabeled data, which is commonly used in classification [4, 19, 28, 42, 45] and scene understanding [7, 29, 36, 37, 44, 57, 60, 61] problems. Only few depth estimation methods [31, 51] adopt the concept of pseudo-labeling. DESC [31] designs a model to predict the instance height and converts the instance height to depth values as the pseudo-label for the depth prediction of the real image. Yang *et al.* [51] generate the pseudo-label from multi-view images and design

(a) Our overall 3D-PL framework for pseudo-labeling.

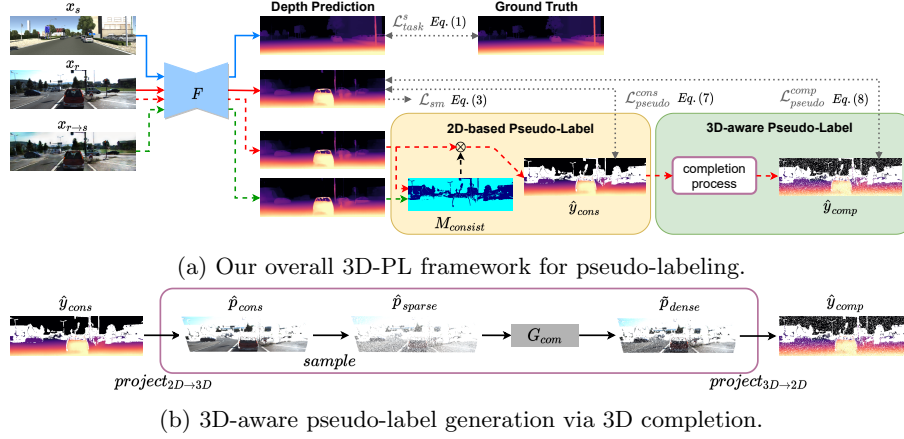(b) 3D-aware pseudo-label generation via 3D completion.

Fig. 2: **(a)** Illustration of our proposed 3D-PL framework together with the training objectives. F is the depth prediction network, with input of the synthetic image x_s, the real image x_r, and the synthetic-stylized image $x_{r \to s}$. In 3D-PL, we obtain 2D-based pseudo-labels $\hat{y}_{cons}$ through finding the region with consistent depth (light blue color in $M_{consist}$) across the predictions of x_r and $x_{r \to s}$ (see Section 3.2), while 3D-aware pseudo-labels $\hat{y}_{comp}$ are obtained via the 3D completion process. Here we denote solid lines as the computation flow where the gradients can be back-propagated, while the dashed lines indicate that pseudo-labels are generated offline based on the preliminary model in Section 3.1. **(b)** We first project the 2D-based pseudo-labels $\hat{y}_{cons}$ to the 3D point cloud $\hat{p}_{cons}$, followed by uniformly sub-sampling $\hat{p}_{cons}$ to sparse $\hat{p}_{sparse}$. Then, the completion network G_{com} densifies $\hat{p}_{sparse}$ to obtain $\tilde{p}_{dense}$, in which we further project $\tilde{p}_{dense}$ back to 2D and produce 3D-aware pseudo-labels $\hat{y}_{comp}$ (see Section 3.2).

a few ways to refine pseudo-labels, including fusing point clouds from multi-views. These methods succeed in producing pseudo-labels, but they require to have the instance information [31] or multi-view images [51]. Moreover, as [51] is a multi-stereo task, it is easier to build a complete point cloud from multi views and render the depth map as pseudo-labels. Their task also focuses on the main object instead of the overall scene. In our method, we design the point cloud completion method to generate reliable 3D-aware pseudo-labels based on a single image that contains a real-world outdoor scene.

3 Proposed Method

Our goal in this paper is to adapt the depth prediction model F to the unlabeled real image x_r as the target domain, where the synthetic image-depth pair (x_s, y_s) in the source domain is provided for supervision. Without domain adaptation, the depth prediction model F can be well trained on the synthetic data (x_s, y_s), but it cannot directly perform well on the real image x_r because of the

domain shift. Thus, we propose our pseudo-labeling method to provide direct supervisions on target image x_r, which reduces the domain gap effectively.

Fig. 2 illustrates the overall pipeline of our method. To utilize our pseudo-labeling techniques, we first use the synthetic data to train a preliminary depth prediction model F, and then adopt this pretrained model to infer pseudo-labels on real data for self-training. For pseudo-label generation, we propose 2D-based and 3D-aware schemes, where we name them as *consistency label* and *completion label*, respectively. We detail our model designs in the following sections.

3.1   Preliminary Model Objectives

Here, we describe the preliminary objectives during our model pre-training by using the synthetic image-depth pairs (x_s, y_s) and the real image x_r, including depth estimation loss and smoothness loss. Please note that, this is a common step before pseudo-labeling, in order to account for initially noisy predictions.
Depth Estimation Loss. As synthetic image-depth pairs (x_s, y_s) can provide the supervision, we directly minimize the L_1 distance between the predicted depth $\tilde{y}_s = F(x_s)$ of the synthetic image x_s and the ground truth depth y_s.

$$\mathcal{L}^s_{task}(F) = \|\tilde{y}_s - y_s\|_1. \tag{1}$$

In addition to the synthetic images x_s, we follow the similar style translation strategy as [58] to generate real-stylized images $x_{s \to r}$, in which $x_{s \to r}$ maintains the content of x_s but has the style from a randomly chosen real image x_r. Note that, to keep the simplicity of our framework, we adopt the real-time style transfer AdaIN [20] (pretrained model provided by [20]) instead of training another translation network like [58].

$$\mathcal{L}^{s \to r}_{task}(F) = \|\tilde{y}_{s \to r} - y_s\|_1. \tag{2}$$

Smoothness Loss. For the target image x_r, we adopt the smoothness loss as [16,58] to encourage the local depth prediction $\tilde{y}_r$ being smooth and consistent. Since depth values are often discontinuous on the boundaries of objects, we weigh this loss with the edge-aware term:

$$\mathcal{L}_{sm}(F) = e^{-\nabla x_r}\|\nabla \tilde{y}_r\|_1, \tag{3}$$

where ∇ is is the first derivative along spatial directions.

3.2   Pseudo-Label Generation

With the preliminary loss functions introduced in Section 3.1 that pre-train the model, we then perform our pseudo-labeling process with two schemes. First, 2D-based consistency label aims to find the highly confident pixels from depth predictions as pseudo-labels. Second, 3D-aware completion label utilizes a 3D completion model G_{com} to refine some prior pseudo-labels and further extend the range of pseudo-labels (see Fig. 2).

2D-based Consistency Label A typical way to discover reliable pseudo-labels is to find confident ones, e.g., utilizing the softmax output from tasks like semantic segmentation [29]. However, due to the nature of the regression task in depth estimation, it is not trivial to obtain such 2D-based pseudo-labels from the network output. Therefore, we design a simple yet effective way to construct the confidence map via feeding the model two target images with the same content but different styles. Since pixels in two images have correspondence, our motivation is that, pixels that are more confident should have more consistent depth values across two predictions (i.e., finding pixels that are more domain invariant through the consistency of predictions from real images with different styles).

To achieve this, we first obtain the synthetic-stylized image $x_{r \to s}$ for the real image x_r, which combines the content of x_r and the style of a synthetic image x_s, via AdaIN [20]. Then, we obtain depth predictions of these two images, $\tilde{y}_r = F(x_r)$, $\tilde{y}_{r \to s} = F(x_{r \to s})$, and calculate their difference. If the difference at one pixel is less then a threshold τ, we consider this pixel as a more confident prediction to form the pseudo-label $\hat{y}_{cons}$. The procedure is written as:

$$M_{consist} = |\tilde{y}_r - \tilde{y}_{r \to s}| < \tau,$$
$$\hat{y}_{cons} = M_{consist} \otimes \tilde{y}_r, \tag{4}$$

where $M_{consist}$ is the binary mask for consistency, which records where pixels are consistent. τ is the threshold, set as 0.5 in meter, and $\otimes$ is the element-wise product to filter the prediction $\tilde{y}_r$ of the target image.

3D-aware Completion Label Since depth estimation is a 3D problem, we expand the prior 2D-based pseudo-label $\hat{y}_{cons}$ to obtain more pseudo-labels in the 3D space, so that the pseudo-labeling process can benefit from the learned 3D structure. To this end, based on the 2D consistency label $\hat{y}_{cons}$, we propose a 3D completion process to reason neighboring relationships in 3D. As shown in Fig. 2b, the 3D completion process adopts the point cloud completion technique to learn from the 3D structure and generate neighboring points.

First, we project the 2D-based pseudo-label $\hat{y}_{cons}$ to point clouds $\hat{p}_{cons} = project_{2D \to 3D}(\hat{y}_{cons})$ in the 3D space. In the projection procedure, we reconstruct each point (x_i, y_i, z_i) from the image pixel (u_i, v_i) with its depth value d_i based on the standard pinhole camera model (more details and discussions are provided in the supplementary material). Next, we uniformly sample points from $\hat{p}_{cons}$ to have sparse point clouds $\hat{p}_{sparse} = sample(\hat{p}_{cons})$, followed by taking $\hat{p}_{sparse}$ as the input to the 3D completion model G_{com} for synthesizing the missing points. Those generated points from the 3D completion model G_{com} compose new dense point clouds $\tilde{p}_{dense} = G_{com}(\hat{p}_{sparse})$, and then we project each point $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ back to the original 2D plane as $(\tilde{u}_i, \tilde{v}_i)$ with updated depth value $\tilde{d}_i = \tilde{z}_i$.

Therefore, our 3D-aware pseudo-label $\hat{y}_{comp}$ (i.e., completion label) is formed by the updated depth value $\tilde{d}_i$. Note that, as there could exist some projected

Input image x_r Ground truth y_r $\hat{y}_{cons}$ $\hat{y}_{comp}$ $\hat{y}_{comp} - \hat{y}_{cons}$
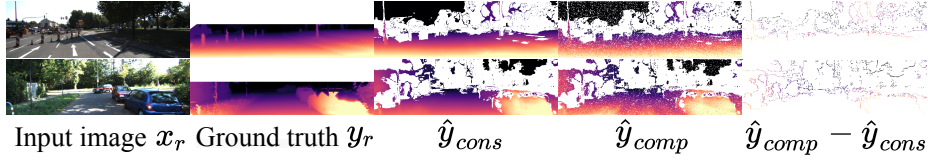
Fig. 3: Examples for our pseudo-labels. The third and fourth columns are pseudo-labels for 2D-based $\hat{y}_{cons}$ and 3D-aware $\hat{y}_{comp}$. The final column represents the complementary pseudo-labels produced by $\hat{y}_{comp}$. Note that ground truth y_r is the reference but not used in our model training.

points falling outside the image plane and not all the pixels on the image plane are covered by the projected points, we construct a mask M_{valid} which records the pixels on the completion label $\hat{y}_{comp}$ where the projection succeeds, i.e., having valid $(\tilde{u}_i, \tilde{v}_i)$.

$$\hat{y}_{comp} = M_{valid} \otimes project_{3D \to 2D}(\tilde{p}_{dense}). \tag{5}$$

In Fig. 3, we show that the 3D-aware completion label $\hat{y}_{comp}$ expands the pseudo-labels from the 2D-based consistency label $\hat{y}_{cons}$, i.e., visualizations in $\hat{y}_{comp} - \hat{y}_{cons}$ are additional pseudo-labels from the 3D completion process (please refer to Section 4.3 for further analyzing the effectiveness of 3D-aware pseudo-labels). **3D Completion Model.** We pre-train the 3D completion model G_{com} using the synthetic ground truth depth y_s in advance and keep it fixed during our completion process. We project the entire y_s to 3D point clouds $\hat{p}^s = project_{2D \to 3D}(y_s)$ and then perform the same process (i.e., sampling and completion) as in Fig. 2b to obtain the generated dense point clouds $\tilde{p}^s_{dense}$. Since $\hat{p}^s$ is the ground truth point clouds of $\tilde{p}^s_{dense}$, we can directly minimize Chamfer Distance (CD) [11] between these two point clouds to train the 3D completion model G_{com}, $\mathcal{L}_{cd}(G_{com}) = CD(\hat{p}^s, \tilde{p}^s_{dense})$.

3.3   Overall Training Pipeline and Objectives

There are two training stages in our proposed method: the first stage is to train a preliminary depth model F, and the second stage is to apply the proposed pseudo-labeling techniques through this preliminary model. The loss in the first stage consists of the ones introduced in Section 3.1:

$$\mathcal{L}_{base} = \lambda_{task}(\mathcal{L}^s_{task} + \mathcal{L}^{s \to r}_{task}) + \lambda_{sm}\mathcal{L}_{sm}, \tag{6}$$

where λ_{task} and λ_{sm} are set as 100 and 0.1 respectively, following the similar settings in [58]. Note that in our implementation, for every synthetic image x_s, we augment three corresponding real-stylized images $x_{s \to r}$, where their styles are obtained from three real images randomly drawn from the training set. **Training with Pseudo-labels.** In the second stage, we use our generated 2D-based and 3D-aware pseudo-labels in Eq. (4) and Eq. (5) to provide direct supervisions on the target image x_r. Since the completion label $\hat{y}_{comp}$ is aware of the

3D structural information and can refine the prior 2D-based pseudo-labels $\hat{y}_{cons}$, we choose the completion label $\hat{y}_{comp}$ as the main reference if a pixel has both consistency label $\hat{y}_{cons}$ and completion label $\hat{y}_{comp}$. The 2D and 3D pseudo-label loss functions are respectively defined as:

$$\mathcal{L}_{pseudo}^{cons}(F) = \|M'_{valid} \otimes (M_{consist} \otimes \tilde{y}_r - \hat{y}_{cons})\|_1, \tag{7}$$

$$\mathcal{L}_{pseudo}^{comp}(F) = \|M_{valid} \otimes \tilde{y}_r - \hat{y}_{comp}\|_1, \tag{8}$$

where $M'_{valid} = (1 - M_{valid})$ is the inverse mask of M_{valid}. In addition to the two pseudo-labeling objectives, we also include the supervised synthetic data to maintain the training stability. The total objective of the second stage is:

$$\begin{aligned}
\mathcal{L}_{total} = {}& \alpha(\lambda_{cons}\mathcal{L}_{pseudo}^{cons} + \lambda_{comp}\mathcal{L}_{pseudo}^{comp}) \\
& +(1-\alpha)\lambda_{task}^s\mathcal{L}_{task}^s + \lambda_{sm}\mathcal{L}_{sm},
\end{aligned} \tag{9}$$

where α set as 0.7 is the proportion ratio between the supervised loss of the synthetic and real image. λ_{task}^s, λ_{cons}, λ_{comp}, and λ_{sm} are set as 100, 1, 0.1, and 0.1, respectively. Here we do not include the $\mathcal{L}_{task}^{s \to r}$ loss as in Eq. (6) to make the model training more focused on the real-domain data.

Stereo Setting. The training strategy mentioned above is under the condition that we can only access the monocular single image of the real data x_r. In addition, if the stereo pairs are available during training as the setting in GASDA [56], we can further include the geometry consistency loss $\mathcal{L}_{tgc}$ in [56] to our proposed method (more details are in the supplementary material):

$$\mathcal{L}_{stereo} = \mathcal{L}_{total} + \lambda_{tgc}\mathcal{L}_{tgc}, \tag{10}$$

where $\mathcal{L}_{total}$ is the loss in Eq. (9), and λ_{tgc} is set as 50 following [56].

4 Experimental Results

In summary, we conduct experiments for the synthetic-to-real benchmark when only single images or stereo pairs are available during training. Then we show ablation studies to demonstrate the effectiveness of the proposed pseudo-labeling methods. Moreover, we provide discussion to validate the effectiveness of our 3D-aware pseudo-labeling method. Finally, we directly evaluate our models on two real-world datasets to show the generalization ability. More results and analysis are provided in the supplementary material.

Datasets and Evaluation Metrics. We adopt Virtual KITTI (vKITTI) [13] and real KITTI [15] as our source and target datasets respectively. vKITTI contains $21,260$ synthetic image-depth pairs of the urban scene under different weather conditions. Since the maximum depth ground truth values are different in vKITTI and KITTI, we clip the maximum value to $80m$ as [58]. For evaluating the generalization ability, we use the KITTI Stereo [34] and Make3D [43] datasets following the prior work [56]. We use the same depth evaluation metrics as [56, 58], including four types of errors and three types of accuracy metrics.

Table 1: Quantitative results on KITTI in the single-image setting, where we denote the best results in bold. For the training data, "K", "CS", and "S" indicate KITTI [15], CityScapes [8], and virtual-KITTI [13] datasets respectively. We highlight the rows in gray for those methods using the domain adaptation (DA) techniques.

Method	Supervised	Dataset	Cap	Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
				Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen et al. [9]	Yes	K	$80m$	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [30]	Yes	K	$80m$	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Zhou et al. [59]	No	K	$80m$	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou et al. [59]	No	K+CS	$80m$	0.198	1.836	6.565	0.275	0.718	0.901	0.960
All synthetic	No	S	$80m$	0.253	2.303	6.953	0.328	0.635	0.856	0.937
All real	No	K	$80m$	0.158	1.151	5.285	0.238	0.811	0.934	0.970
AdaDepth [25]	No	K+S(DA)	$80m$	0.214	1.932	7.157	0.295	0.665	0.882	0.950
T^2Net [58]	No	K+S(DA)	$80m$	0.182	1.611	6.216	0.265	0.749	0.898	0.959
3D-PL (Ours)	No	K+S(DA)	$80m$	**0.169**	**1.371**	**6.037**	**0.256**	**0.759**	**0.904**	**0.961**
Garg et al. [14]	No	K	$50m$	0.169	1.080	5.104	0.273	0.740	0.904	0.962
All synthetic	No	S	$50m$	0.244	1.771	5.354	0.313	0.647	0.866	0.943
All real	No	K	$50m$	0.151	0.856	4.043	0.227	0.824	0.940	0.973
AdaDepth [25]	No	K+S(DA)	$50m$	0.203	1.734	6.251	0.284	0.687	0.899	0.958
T^2Net [58]	No	K+S(DA)	$50m$	0.168	1.199	4.674	0.243	0.772	0.912	0.966
3D-PL (Ours)	No	K+S(DA)	$50m$	**0.162**	**1.049**	**4.463**	**0.239**	**0.776**	**0.916**	**0.968**

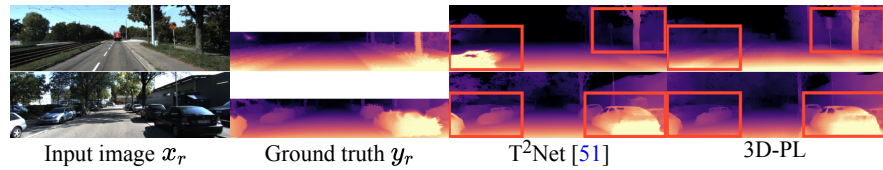Input image x_r Ground truth y_r T^2Net [51]   3D-PL

Fig. 4: Qualitative results on KITTI in the single-image setting. We show that our 3D-PL produces more accurate results for the tree and grass (upper row) and better shapes in the car (bottom row), compared to the T^2Net [58] method.

Implementation Details. Our depth prediction model F adopts the same U-net [41] structure as [58]. Following [50], the 3D completion model G_{com} is modified from PCN [53] with PointNet [39]. We implement our model based on the Pytorch framework with NVIDIA Geforce GTX 2080 Ti GPU. All networks are trained with the Adam optimizer. The depth prediction model F and 3D completion model G_{com} are trained from scratch with learning rate 10^{-4} and linear decay after 10 epochs. We train F for 20 epochs in the first stage and 10 epochs in the second stage, and pre-train G_{com} for 20 epochs. The style transfer network AdaIN is pre-trained without any finetuning.

4.1 Synthetic-to-Real Benchmark

We follow [58] to use $22,600$ KITTI images from 32 scenes as the real training data, and evaluate the performance on the eigen test split [9] of 697 images from other 29 scenes. Following [56], we evaluate the depth prediction results with the ground truth depth less than $80m$ or $50m$. There are two real-data training settings in domain adaptation for monocular depth estimation: 1) only single real images are available and we cannot access binocular or semantic

Table 2: Quantitative results on KITTI with having stereo pairs during training.

Method	Cap	Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Synthetic + Stereo	$80m$	0.151	1.176	5.496	0.237	0.787	0.926	0.972
T^2Net [58] + Stereo	$80m$	0.154	1.115	5.504	0.233	0.800	0.929	0.971
GASDA [56] (Stereo)	$80m$	0.149	1.003	4.995	0.227	0.824	0.941	0.973
DESC [31] + Stereo	$80m$	0.122	0.946	5.019	0.217	0.843	0.942	0.974
SharinGAN [38] (Stereo)	$80m$	0.116	0.939	5.068	0.203	0.850	0.948	0.978
3D-PL + Stereo	$80m$	**0.113**	**0.903**	**4.902**	**0.201**	**0.859**	**0.952**	**0.979**
Synthetic + Stereo	$50m$	0.145	0.909	4.204	0.224	0.800	0.934	0.975
T^2Net [58] + Stereo	$50m$	0.148	0.828	4.123	0.219	0.815	0.938	0.975
GASDA [56] (Stereo)	$50m$	0.143	0.756	3.846	0.217	0.836	0.946	0.976
DESC [31] + Stereo	$50m$	0.116	0.725	3.880	0.206	0.855	0.948	0.976
SharinGAN [38] (Stereo)	$50m$	0.109	0.673	3.770	0.190	0.864	0.954	0.981
3D-PL + Stereo	$50m$	**0.106**	**0.641**	**3.643**	**0.189**	**0.872**	**0.958**	**0.982**

information as [58]; 2) stereo pairs are available during training, so that geometry consistency can be leveraged as [56]. Our pseudo-labeling method does not have an assumption of the data requirement, and hence we conduct experiments in these two different data settings as mentioned in Section 3.3.

Single-image Setting. In this setting, we can only access monocular real images in the whole training process, as the overall objective in Eq. (9). Table 1 shows the quantitative results, where the domain adaptation methods are highlighted in gray. "All synthetic/All real" are only trained on synthetic/real image-depth pairs, which can be viewed as the lower/upper bound. Our 3D-PL method outperforms T^2Net (state-of-the-art) in every metric, especially 13% and 15% improvement in the "Sq Rel" error of $50m$ and $80m$. Fig. 4 shows the qualitative results, where we compare our 3D-PL with T^2Net [58]. In the upper row, 3D-PL produces more accurate results for the tree and grass, while T^2Net predicts too far and close respectively. In the lower row, our result has a better shape in the right car and more precise depth for the left two cars.

Stereo-pair Setting. If stereo pairs are available, we can utilize the geometry constraints to have self-supervised stereo supervisions as [56] using the objective in Eq. (10). Table 2 shows that our 3D-PL achieves the best performance among state-of-the-art methods. In particular, without utilizing any other clues from real-world semantic annotation, 3D-PL outperforms DESC [31] with 12% lower "Sq Rel" error in the stereo scenario. This shows that our pseudo-labeling is able to generate more reliable pseudo-labels over the single-image setting.

Fig. 5 shows qualitative results, where we compare our 3D-PL with DESC [31] + Stereo and SharinGAN [38]. 3D-PL produces better results on the overall structure (e.g., tree, wall, and car in the top row). For challenging situations such as closer objects standing alone and hiding in a complicated farther background (e.g., road sign in the middle row, tree in the bottom row), other methods tend to produce similar depth values as the background, while 3D-PL predicts better object shape and distinguish the object from the background even if it is very thin. (e.g., traffic light in the bottom row). This shows the benefits of our 3D-aware pseudo-labeling design, which reasons the 3D structural information.

Table 3: Ablation study on KITTI in the single-image setting.

Method	Error Metrics (lower, better)			
	Abs Rel	Sq Rel	RMSE	RMSE log
Only synthetic	0.244	1.771	5.354	0.313
$+\hat{y}_{cons}$ (all pixels)	0.166	1.125	4.557	0.244
$+\hat{y}_{cons}$ (confident)	0.163	1.095	4.555	0.243
$+\hat{y}_{comp}$ (confident)	0.164	1.054	4.473	**0.239**
3D-PL ($\hat{y}_{comp}$ all pixels)	**0.161**	1.070	4.504	0.240
3D-PL ($\hat{y}_{comp}$ confident)	0.162	**1.049**	**4.463**	**0.239**

Input image x_r Ground truth y_r DESC [29]   SharinGAN[33]   3D-PL+Stereo
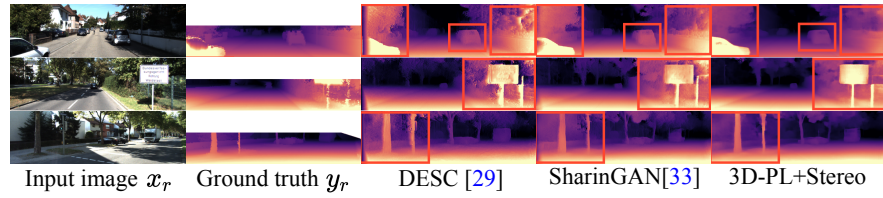
Fig. 5: Qualitative results on KITTI with having stereo pairs during training. We show that our 3D-PL produces better results on the overall structure (e.g., tree, wall, and car in the top row), closer objects (e.g., road sign in the middle row, tree in the bottom row), and shapes (e.g., traffic light in the bottom row), compared to DESC [31] and SharinGAN [38].

4.2   Ablation Study

We demonstrate the contributions of our model designs in Table 3 using the "50m Cap" and single-image settings, where "Only synthetic" trains only on the supervised synthetic image-depth pairs.

Importance of Pseudo-labels. First, we show that either using the 2D-based or 3D-aware pseudo-labels improve the performance, i.e., "$+\hat{y}_{cons}$ (confident)" and "$+\hat{y}_{comp}$ (confident)". Then, our final model in "3D-PL ($\hat{y}_{comp}$ confident)" further improves depth estimation, and shows the complementary properties of using both 2D-based and 3D-aware pseudo-labels.

Importance of Consistency Mask. We show the importance of having the consistency mask in Eq. (4) as the confidence measure. For the 2D-based pseudo-label, we compare the result of using the consistency mask "$+\hat{y}_{cons}$ (confident)" and the one using the entire depth prediction as the pseudo-label, "$+\hat{y}_{cons}$ (all pixels)". With the consistency mask, it has 3% lower in the "Sq Rel" error. Moreover, this consistency mask also improves 3D-aware pseudo-labeling when we project depth values to point clouds for 3D completion. When inputting all the pixels for this process, i.e., "3D-PL ($\hat{y}_{comp}$ all pixels)", this may include less accurate depth values for performing 3D completion, which results in less reliable pseudo-labels compared to our final model using the confident pixels, i.e., "3D-PL ($\hat{y}_{comp}$ confident)".

Table 4: Statistics of pixel proportion in our 2D/3D pseudo-labels. "R" and "E" indicate "refined" and "extended".

Method	2D Proportion	3D Proportion
2D only $(+\hat{y}_{cons})$	48.91%	0%
3D-PL	5.28%	43.63% (R) + 3.9% (E)

Table 5: Results of using either the 2D-based $\hat{y}_{cons}$ or the 3D-aware $\hat{y}_{comp}$ pseudo-label as the reference, when there is a duplication on both pseudo-labels.

Main Reference	Error Metrics (lower, better)			
	Abs Rel	Sq Rel	RMSE	RMSE log
Completion Label $\hat{y}_{comp}$	**0.162**	**1.049**	**4.463**	**0.239**
Consistency Label $\hat{y}_{cons}$	0.164	1.095	4.529	0.243

4.3   Effectiveness of 3D-aware Pseudo-labels

To show the impact of 3D-aware pseudo-labels, we compute the proportions of pixels chosen as 2D/3D pseudo-labels in each image and take the average as the final statistics. The effectiveness of 3D-aware pseudo-labels is in two-fold: **refine** and **extend** from 2D-based pseudo-labels. In Table 4, the initial proportion of confident 2D-based pseudo-labels "2D only $(+\hat{y}_{cons})$" is 48.91% among image pixels. As stated in Section 3.3, 3D-PL improves original 2D labels, which results in 43.63% refined and 3.9% extended labels. The rightmost subfigure of Fig. 3 visualizes extended labels $\hat{y}_{comp} - \hat{y}_{cons}$, in which it shows that the improved performance is contributed by the larger proportion of 3D-aware pseudo-labels. **Ability of pseudo-label refinement.** Since the 2D-based and 3D-aware pseudo-labels may have the duplication on the same pixel, we conduct experiments to use either $\hat{y}_{cons}$ or $\hat{y}_{comp}$ as the reference when such cases happen. In Table 5, choosing $\hat{y}_{comp}$ as the main reference has the better performance, which indicates that updating the pseudo-label of a pixel from original $\hat{y}_{cons}$ to $\hat{y}_{comp}$ brings the positive effect. This validates that $\hat{y}_{comp}$ can refine the prior 2D-based pseudo-labels since it is aware of the 3D structural information.

4.4   Generalization to Real-world Datasets

KITTI Stereo. We evaluate our model on 200 images of KITTI stereo 2015 [34], which is a small subset of KITTI images but has different ways of collecting groundtruth of depth information. Since the ground truth of KITTI stereo has been optimized for the moving objects, it is denser than LiDAR, especially for the vehicles. Note that, this benefits DESC [31] in this evaluation as their method relies on the instance information from the pre-trained segmentation model. Table 6 shows the quantitative results, where our 3D-PL in both single-image and stereo settings performs competitively against existing methods.
Make3D Dataset. Moreover, we directly evaluate the model on the Make3D dataset [43] without any finetuning. We choose 134 test images with central image crop and clamp the depth value to $70m$, following [16]. Here, since Make3D

Table 6: Quantitative results on KITTI stereo 2015 benchmark [34]. "S$^\diamond$" denotes synthetic data that [2] captures from GTA [40]. "Supervised" represents whether the method is trained on KITTI stereo.

Method	Supervised	Dataset	Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
			Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Godard *et al.* [16]	No	K	0.124	1.388	6.125	0.217	0.841	0.936	0.975
Godard *et al.* [16]	No	K+CS	0.104	1.070	5.417	0.188	0.875	0.956	0.983
Atapour *et al.* [2]	No	K+S$^\diamond$(DA)	0.101	1.048	5.308	0.184	0.903	0.988	0.992
T^{2}Net [58]	No	K+S(DA)	0.155	1.731	6.510	0.237	**0.800**	**0.921**	**0.969**
3D-PL	No	K+S(DA)	**0.147**	**1.352**	**6.157**	**0.233**	**0.800**	0.918	0.967
GASDA [56] (Stereo)	No	K+S(DA)	0.106	0.987	5.215	0.176	0.885	0.963	0.986
DESC [31] + Stereo	No	K+S(DA)	**0.085**	**0.781**	4.490	0.158	0.909	0.967	0.986
SharinGAN [38] (Stereo)	No	K+S(DA)	0.092	0.904	4.614	0.159	0.906	0.969	0.987
3D-PL + Stereo	No	K+S(DA)	**0.085**	0.830	**4.489**	**0.149**	**0.915**	**0.971**	**0.988**

Table 7: Quantitative results on Make3D [43]. "Supervised" represents whether the method is trained on Make3D.

Method	Supervised	Error Metrics (lower, better)		
		Abs Rel	Sq Rel	RMSE
Karsch *et al.* [24]	Yes	0.398	4.723	7.801
Laina *et al.* [27]	Yes	0.198	1.665	5.461
AdaDepth [25]	Yes	0.452	5.71	9.559
Godard *et al.* [16]	No	0.505	10.172	10.936
AdaDepth [25]	No	0.647	12.341	11.567
T^{2}Net [58]	No	0.508	6.589	8.935
Atapour *et al.* [2]	No	0.423	9.343	9.002
GASDA [56]	No	0.403	6.709	10.424
DESC [31]	No	0.393	4.604	8.126
SharinGAN [38]	No	0.377	4.900	8.388
S2R-DepthNet [5]	No	0.490	10.681	10.892
3D-PL	No	**0.352**	**3.539**	**7.967**

is a different domain from the KITTI training data, we apply the single-image model to reduce the strong domain-related constraints such as the stereo supervisions. In Table 7, 3D-PL achieves the best performance compared to other approaches. It is also worth mentioning that 3D-PL outperforms the domain generalization method [5] and supervised method [24] by 66% and 25% in "Sq Rel", showing the promising generalization capability.

5 Conclusions

In this paper, we introduce a domain adaptation method for monocular depth estimation. We propose 2D-based and 3D-aware pseudo-labeling mechanisms, which utilize knowledge from synthetic domain as well as 3D structural information to generate reliable pseudo depth labels for real data. Extensive experiments show that our pseudo-labeling strategies are able to improve depth estimation in various settings against several state-of-the-art domain adaptation approaches, as well as achieving good performance in unseen datasets for generalization.

References

1. Amiri, A.J., Loo, S.Y., Zhang, H.: Semi-supervised monocular depth estimation with left-right consistency using deep neural network. In: IEEE International Conference on Robotics and Biomimetics (ROBIO) (2019) 4

2. Atapour-Abarghouei, A., Breckon, T.P.: Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4, 14

3. Cao, Y., Wu, Z., Shen, C.: Estimating depth from monocular images as classification using deep fully convolutional residual networks. IEEE Transactions on Circuits and Systems for Video Technology (TCSVT) (2017) 1, 3

4. Chen, C., Xie, W., Huang, W., Rong, Y., Ding, X., Huang, Y., Xu, T., Huang, J.: Progressive feature alignment for unsupervised domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 4

5. Chen, X., Wang, Y., Chen, X., Zeng, W.: S2r-depthnet: Learning a generalizable depth-specific structural representation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 14

6. Chen, Y.C., Lin, Y.Y., Yang, M.H., Huang, J.B.: Crdoco: Pixel-level domain transfer with cross-domain consistency. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2, 4

7. Chen, Z., Zhang, R., Zhang, G., Ma, Z., Lei, T.: Digging into pseudo label: a low-budget approach for semi-supervised semantic segmentation. IEEE Access (2020) 4

8. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 10

9. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. ArXiv:1406.2283 (2014) 1, 3, 10

10. Eldesokey, A., Felsberg, M., Holmquist, K., Persson, M.: Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 22, 23

11. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 8

12. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 1, 3

13. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 3, 9, 10, 22, 25

14. Garg, R., Bg, V.K., Carneiro, G., Reid, I.: Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision (ECCV) (2016) 1, 4, 10

15. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012) 1, 3, 9, 10, 22, 23, 26, 27

16. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 1, 4, 6, 13, 14, 19

17. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: IEEE International Conference on Computer Vision (ICCV) (2019) 1, 4
18. Guizilini, V., Li, J., Ambrus, R., Pillai, S., Gaidon, A.: Robust semi-supervised monocular depth estimation with reprojected distances. In: Conference on Robot Learning (CoRL) (2020) 4
19. Hu, Z., Yang, Z., Hu, X., Nevatia, R.: Simple: Similar pseudo label exploitation for semi-supervised classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 4
20. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: IEEE International Conference on Computer Vision (ICCV) (2017) 2, 6, 7
21. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in Neural Information Processing Systems (NeurIPS) **28** (2015) 19
22. Ji, R., Li, K., Wang, Y., Sun, X., Guo, F., Guo, X., Wu, Y., Huang, F., Luo, J.: Semi-supervised adversarial monocular depth estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2019) 4
23. Jiao, J., Cao, Y., Song, Y., Lau, R.: Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In: European Conference on Computer Vision (ECCV) (2018) 1, 3
24. Karsch, K., Liu, C., Kang, S.B.: Depth transfer: Depth extraction from video using non-parametric sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2014) 14
25. Kundu, J.N., Uppala, P.K., Pahuja, A., Babu, R.V.: Adadepth: Unsupervised content congruent adaptation for depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 2, 4, 10, 14
26. Kuznietsov, Y., Stuckler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 4
27. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: International Conference on 3D Vision (3DV) (2016) 14
28. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: International Conference on Machine Learning (ICML) (2013) 4
29. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2, 4, 7
30. Liu, F., Shen, C., Lin, G., Reid, I.: Learning depth from single monocular images using deep convolutional neural fields. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2015) 1, 3, 10
31. Lopez-Rodriguez, A., Mikolajczyk, K.: Desc: Domain adaptation for depth estimation via semantic consistency. ArXiv:2009.01579 (2020) 2, 3, 4, 5, 11, 12, 13, 14, 27, 28
32. Ma, F., Cavalheiro, G.V., Karaman, S.: Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In: IEEE International Conference on Robotics and Automation (ICRA) (2019) 22
33. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 1, 4

34. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 9, 13, 14, 26, 27, 28
35. Park, J., Joo, K., Hu, Z., Liu, C.K., So Kweon, I.: Non-local spatial propagation network for depth completion. In: European Conference on Computer Vision (ECCV) (2020) 22
36. Pastore, G., Cermelli, F., Xian, Y., Mancini, M., Akata, Z., Caputo, B.: A closer look at self-training for zero-label semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 4
37. Paul, S., Tsai, Y.H., Schulter, S., Roy-Chowdhury, A.K., Chandraker, M.: Domain adaptive semantic segmentation using weak labels. In: European Conference on Computer Vision (ECCV) (2020) 4
38. PNVR, K., Zhou, H., Jacobs, D.: Sharingan: Combining synthetic and real data for unsupervised geometry estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4, 11, 12, 14, 27, 28
39. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 10, 22
40. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: European Conference on Computer Vision (ECCV) (2016) 14
41. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2015) 10
42. Saito, K., Ushiku, Y., Harada, T.: Asymmetric tri-training for unsupervised domain adaptation. In: International Conference on Machine Learning (ICML) (2017) 4
43. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2008) 3, 9, 13, 14, 26, 28
44. Shin, I., Tsai, Y.H., Zhuang, B., Schulter, S., Liu, B., Garg, S., Kweon, I.S., Yoon, K.J.: Mm-tta: Multi-modal test-time adaptation for 3d semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 4
45. Taherkhani, F., Dabouei, A., Soleymani, S., Dawson, J., Nasrabadi, N.M.: Self-supervised wasserstein pseudo-labeling for semi-supervised image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 4
46. Tosi, F., Aleotti, F., Poggi, M., Mattoccia, S.: Learning monocular depth estimation infusing traditional stereo knowledge. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1, 4
47. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2
48. Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., Yuille, A.L.: Towards unified depth and semantic prediction from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 3
49. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8445–8453 (2019) 24
50. Xiang, R., Zheng, F., Su, H., Zhang, Z.: 3ddepthnet: Point cloud guided depth completion network for sparse depth and single color image. ArXiv:2003.09175 (2020) 10, 22

51. Yang, J., Alvarez, J.M., Liu, M.: Self-supervised learning of depth inference for multi-view stereo. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 4, 5
52. Yin, W., Liu, Y., Shen, C., Yan, Y.: Enforcing geometric constraints of virtual normal for depth prediction. In: IEEE International Conference on Computer Vision (ICCV) (2019) 1, 3
53. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: International Conference on 3D Vision (3DV) (2018) 10, 22
54. Zhan, H., Garg, R., Weerasekera, C.S., Li, K., Agarwal, H., Reid, I.: Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 1, 4
55. Zhang, Z.: Microsoft kinect sensor and its effect. IEEE multimedia (2012) 1
56. Zhao, S., Fu, H., Gong, M., Tao, D.: Geometry-aware symmetric domain adaptation for monocular depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2, 3, 4, 9, 10, 11, 14, 18, 19, 23
57. Zhao, X., Schulter, S., Sharma, G., Tsai, Y.H., Chandraker, M., Wu, Y.: Object detection with a unified label space from multiple datasets. In: European Conference on Computer Vision (ECCV) (2020) 4
58. Zheng, C., Cham, T.J., Cai, J.: T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In: European Conference on Computer Vision (ECCV) (2018) 2, 4, 6, 8, 9, 10, 11, 14, 23, 26, 27, 28
59. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 10
60. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: European Conference on Computer Vision (ECCV) (2018) 2, 4
61. Zou, Y., Zhang, Z., Zhang, H., Li, C.L., Bian, X., Huang, J.B., Pfister, T.: Pseudoseg: Designing pseudo labels for semantic segmentation. ArXiv:2010.09713 (2020) 4

Supplemental Materials

6   Stereo Setting

In this section, we provide the details for objective and the overall training pipeline in the stereo-pair setting as introduced in Section 3.3 of the main paper.

6.1   Objectives

When stereo pairs are available, we utilize the geometry constraints to have self-supervised stereo supervisions as GASDA [56] using the geometry consistency loss. The stereo pairs contain the left image x_r, which is also used in the single-image setting, and the corresponding right image. Here we denote the left and right real images as x^{left}, x^{right} (we ignore the notation r in the real image like x_r^{left} for simplicity) and their depth prediction $\tilde{y}^{left} = F(x^{left}), \tilde{y}^{right} =$

$F(x^{right})$. The geometry consistency loss in GASDA [56] is a reconstruction penalty between the real left image x^{left} and the warped left image $\tilde{x}^{left}$.

$$\mathcal{L}_{tgc}^{left}(F) = \eta\frac{1 - SSIM(x^{left}, \tilde{x}^{left})}{2} + \mu||x^{left} - \tilde{x}^{left}||, \tag{11}$$

where η and μ are set as 0.85 and 0.15 respectively following [56]. The warped left image $\tilde{x}^{left}$ is obtained from the disparity a and the right image x^{right} with bilinear sampling [21] following [16]:

$$\tilde{x}^{left} = x^{right} - a, \tag{12}$$

Since we know the camera parameters when collecting the stereo images, we can convert the depth prediction $\tilde{y}^{left}$ of left image to the disparity a through:

$$a = \frac{b \cdot f}{\tilde{y}^{left}}, \tag{13}$$

where b is the baseline distance between the two cameras and f is the focal length, both parameters are known in the stereo-pair setting. In addition to reconstructing $\tilde{x}^{left}$ from the right image x^{right}, we also warp $\tilde{y}^{right}$ and x^{left} to get $\tilde{x}^{right}$ in our experiments using a similar process and loss $\mathcal{L}_{tgc}^{right}$. Finally, our geometry consistency loss is $\mathcal{L}_{tgc} = \mathcal{L}_{tgc}^{left} + \mathcal{L}_{tgc}^{right}$.

6.2   Overall Training Pipeline

In our stereo-pair setting, there are also two training stages for training a preliminary depth model F and applying the proposed pseudo-labeling techniques through this preliminary model.

Training a preliminary depth model F. In the stereo-pair setting, we follow the single-image setting to use Eq.(6) in the main paper and train a preliminary depth model F for 20 epochs and further train another 10 epochs with adding $\mathcal{L}_{tgc}$:

$$\mathcal{L}_{base}^{stereo} = \lambda_{task}\mathcal{L}_{task}^{s} + \lambda_{sm}\mathcal{L}_{sm} + \lambda_{tgc}\mathcal{L}_{tgc}, \tag{14}$$

where λ_{task}, λ_{sm}, and λ_{tgc} are set as 100, 0.1, and 50 respectively. Here we do not include the $\mathcal{L}_{task}^{s \to r}$ loss to make the model training more focused on the real-domain data. In the second stage, the overall loss is defined as Eq.(10) in the main paper.

7   Sensitivity Analysis

In this section, we analyze the impact of different parameters, such as threshold or the weights in the loss. All experiments are performed in the single-image setting.

7.1   Threshold τ

Table 8 shows our results under different threshold τ as defined in Eq.(4) of the main paper, which controls the range of pseudo-label. The higher threshold means more pseudo-labels are chosen but may not be accurate, while the lower one can obtain more precise pseudo-labels but the amount is less. As shown in Table 8, our method performs robustly under a reasonable range of τ (e.g., 0.3 to 1 meter).

Threshold	Error Metrics (lower, better)			
	Abs Rel	Sq Rel	RMSE	RMSE log
$\tau = 0.1$	**0.161**	1.048	4.497	**0.239**
$\tau = 0.3$	0.162	**1.045**	4.476	**0.239**
$\underline{\tau = 0.5}$	0.162	1.049	**4.463**	**0.239**
$\tau = 1$	**0.161**	1.053	**4.463**	**0.239**
$\tau = 2$	**0.161**	1.060	4.468	**0.239**
$\tau = 3$	0.162	1.066	4.473	**0.239**

Table 8: Our results of different thresholds τ. The unit of τ is meter. Underline denotes our final setting.

7.2   Proportion of Pseudo-label Loss α

Table 9 shows the experiments of using different weight proportion between the pseudo-label loss on real data and the task loss on synthetic data, where α is defined in Eq.(9) of the main paper. With increasing the weight of pseudo-label loss, e.g., $\alpha = 0.3$ to 0.7, the performance is gradually improved, which shows the benefits of our proposed pseudo-labeling strategy. However, the performance drops when α becomes too large, which indicates the importance of having the accurate supervisions from the synthetic data to stabilize model training.

α	$1 - \alpha$	Error Metrics (lower, better)			
		Abs Rel	Sq Rel	RMSE	RMSE log
0.3	0.7	**0.161**	1.051	4.520	0.241
0.5	0.5	**0.161**	1.053	4.490	0.240
$\underline{0.7}$	$\underline{0.3}$	0.162	1.049	**4.463**	**0.239**
0.9	0.1	0.164	**1.043**	4.508	0.240
1	0	0.191	1.175	4.472	0.253

Table 9: Our results of using different proportions α between the pseudo-label loss (α) and the task loss ($1 - \alpha$). Underline denotes our final setting.

7.3   Weighted Terms λ_{cons} and λ_{comp}

Table 10 shows the results of different values of weighted terms (λ_{cons}, λ_{comp}) between 2D-based and 3D-aware pseudo-label loss($\mathcal{L}_{pseudo}^{cons}$, $\mathcal{L}_{pseudo}^{comp}$), defined in Eq.(9) of the main paper. As shown in Table 10, our method performs robustly under a reasonable range of λ_{cons} and λ_{comp} if they do not become too large. We also note that, since the 2D position projected from 3D has a little scale shift to the original 2D pixel on the image plane, there exists scale difference between $\mathcal{L}_{pseudo}^{cons}(\approx 10^{-3})$ and $\mathcal{L}_{pseudo}^{comp}(\approx 10^{-2})$. Thus, we use 10 times λ_{cons} than λ_{comp} as our final setting.

λ_{cons}	λ_{comp}	Error Metrics (lower, better)			
		Abs Rel	Sq Rel	RMSE	RMSE log
0.1	0.01	**0.162**	1.062	4.711	0.247
0.1	0.1	0.163	**1.045**	4.483	0.240
<u>1</u>	<u>0.1</u>	**0.162**	1.049	**4.463**	**0.239**
1	1	0.169	1.097	**4.463**	0.243
10	1	0.168	1.102	4.485	0.243
10	10	0.171	1.138	4.504	0.244

Table 10: Our results of using different values of weighted term ($\lambda_{cons}, \lambda_{comp}$) between 2D and 3D pseudo-label loss. Underline denotes our final setting.

8   More Experiments

In this section, we provide experiments for showing the effectiveness of 3D completion model G_{com}, the comparison with 2D depth completion model, the design choices of the depth estimation loss $\mathcal{L}_{task}$, and the model complexity.

8.1   Effectiveness of G_{com}

We verify whether the 3D completion model G_{com} is well trained. To this end, we simply take one sequence "0018" out of Virtual KITTI dataset as the testing set while the remaining is the training set, and then use the same training procedure stated in the main paper to train our 3D completion model G_{com}. During evaluation, we first project the 2D ground truth depth y_s in testing set to 3D point clouds and uniformly sample them to have sparse point cloud $\hat{p}_{sparse}^s$, and then we take $\hat{p}_{sparse}^s$ as the input of G_{com} to obtain the result of completion $\tilde{p}_{dense}^s$. Finally, we project $\tilde{p}_{dense}^s$ to the 2D depth map $\tilde{y}_{dense}^s$ and measure the depth accuracy with its original ground truth y_s. Table 11 shows that G_{com} has the ability to produce precise and reasonable 3D completion results.

We also provide details for network architecture and sampling strategy of our completion model G_{com}.

Method	Accuracy Metrics (higher, better)		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
3D completion model G_{com}	0.976	0.991	0.995

Table 11: Performance of the completion model G_{com} trained on the synthetic dataset.

Network Architecture of G_{com}. 3D completion model G_{com} is modified from PCN [53]. We follows [50] to adjust the PCN network, including the encoder and the decoder. The encoder of our completion model G_{com} is simplified to one PointNet [39] layer. Our decoder only uses the second stage of point generation in PCN, and we take our sparse point cloud as the "Coarse Output" in PCN. The whole network architecture will be made available to the public.

Sampling Strategy. We "uniformly" sample 3D point cloud into 30720 (25% of pixel number in an image) sparse points as the input to the 3D completion model G_{com}. The 3D point cloud before sampling is projected from 2D depth map through the projection mechanism introduced in Section 9.1. During the pre-training process of G_{com}, we sample the point cloud projected from synthetic ground truth depth y_s to sparse point cloud $\hat{p}^s_{sparse}$. In 3D-aware pseudo-labeling generation, we project 2D pseudo-labels to 3D as point clouds $\hat{p}_{cons}$ and sample $\hat{p}_{cons}$ to sparse point cloud $\hat{p}_{sparse}$.

8.2   Comparison with 2D Depth Completion Model

Since there exists 2D depth completion methods which are also able to complete depth values directly on 2D depth-maps/image [10, 32, 35], we compare our 3D point cloud completion model G_{com} with a 2D depth completion model [10] to validate the necessity of our 3D-aware approach. We apply a recent 2D depth completion model [10] to the sparse depth map sampled from our confident area with two types of training setting. One is pre-trained model provided by the author, and the other one is the model trained from scratch on vKITTI [13] with the same setting as our completion model G_{com}.

Note that our 3D completion model G_{com} is only trained on vKITTI [13] and the 2D depth completion model provided by the author is pre-trained on KITTI [15], so the 2D depth completion model accesses more information from the real domain. We replace our 3D completion model G_{com} with the 2D depth completion model [10] to generate pseudo-labels for training depth prediction model F. As shown in Table 12, even "+ 2D depth completion [10](pre-trained by authors)" is pre-trained on KITTI supervisedly (i.e., using the ground truth depths for training), our proposed 3D-aware approach (i.e., "+ 3D-aware completion label $\hat{y}_{cons}$") provides better performance in all metrics. In addition, we re-train the 2D depth completion model [10] with the same training setting as ours (i.e., trained on vKITTI), and our proposed 3D-aware approach reaches 29% lower error on the "Sq Rel" metric. This shows that our 3D completion model G_{com}, which explicitly considers the 3D structural information, is able to produce more reliable pseudo-labels than the 2D depth completion models.

Method	Abs Rel	Sq Rel	RMSE	RMSE log
+ 2D depth completion [10] (pre-trained by authors)	**0.164**	1.068	4.746	0.247
+ 2D depth completion [10] (trained on vKITTI)	0.186	1.476	6.125	0.310
+ 3D-aware completion label $\hat{y}_{cons}$ (Ours)	**0.164**	**1.054**	**4.473**	**0.239**

Table 12: Training depth prediction model F by using the pseudo-labels generated from different completion models. Note that "+ 2D depth completion [10] (pre-trained by authors)" is pre-trained on KITTI [15], in which the 2D depth completion model [10] has the supervision on depth directly from the real domain, while our model is trained on vKITTI.

8.3   Design Choices of Depth Estimation Loss $\mathcal{L}_{task}$

As stated in Section 7.2, when training with pseudo-labels, it is important to have accurate supervisions on the depth estimation loss $\mathcal{L}_{task}$ to stabilize model training. In Eq.(9) of the main paper, we retain $\mathcal{L}_{task}^{s}$ as the depth estimation to make the model training more focused on the real-domain data. While there exists another option $\mathcal{L}_{task}^{s \to r}$ for the depth estimation loss, as $\mathcal{L}_{task}^{s \to r}$ considers real-stylized images, images produced by style transfer may not align with their original depth ground truths well. Table 13 shows the experiments of adopting different options for the depth estimation loss in Eq.(9) of the main paper, which demonstrates that using $\mathcal{L}_{task}^{s}$ instead of $\mathcal{L}_{task}^{s \to r}$ has lower errors.

Method	Abs Rel	Sq Rel	RMSE	RMSE log
using both $\mathcal{L}_{task}^{s}$ and $\mathcal{L}_{task}^{s \to r}$	**0.160**	1.074	4.611	0.246
using $\mathcal{L}_{task}^{s \to r}$ only	0.161	1.090	4.635	0.247
using $\mathcal{L}_{task}^{s}$ only	0.162	**1.049**	**4.463**	**0.239**

Table 13: Different options for the depth estimation loss $\mathcal{L}_{task}$ in Eq.(9) of the main paper.

8.4   Model Complexity

We analyze the model complexity by computing the number of parameters and the training/testing time for our models. We have depth prediction model F and the completion model G_{com}. The completion model G_{com} only contains 1.324M parameters, which is much smaller than the depth model F (54.565M). The training time for depth model F and completion model G_{com} are 80 and 21 hours. During testing, only the depth model F is required, where it does not introduce additional overheads compared to normal model inference (0.014 seconds for a 192×640 image as used in [56, 58]).

8.5   Application on 3D Object Detection

To show the effectiveness of our depth result, we apply the final depth prediction to the 3D object detection task. We adopt Pseudo-LiDAR [49] to convert our generated depth map to pseudo LiDAR, and take the pseudo LiDAR as the input to the 3D object detection model. We show example results in Figure 6 compared to the ground truths. We also follow [49] to evaluate the result on the validation set of KITTI object detection benchmark for the "car" category. With the IoU threshold at 0.7, the average precision for the 3D object box detection (AP_{3D}) is 15.8%, 12.3%, and 11.2% for easy, moderate, and hard cases, respectively.

Input image   Ground truth   3D-PL + Pseudo-LiDAR [13]

Fig. 6: 3D object detection results using 3D-PL.

9   Details for 2D/3D Projection

We provide the implementation details for the projection procedure between 2D and 3D, including the projection mechanism and some discussions.

9.1   Projection Mechanism

Projection from 2D to 3D. We aim to reconstruct each point (x_i, y_i, z_i) in the 3D space from the 2D image pixel (u_i, v_i) with its depth value d_i based on the standard pinhole camera model. We assume the size of image is $H \times W$ and the pixel positions on the original image plane are $\{(u_i, v_i)\}_{i=1}^{H \times W}$, where each pixel (u_i, v_i) has the corresponding depth value d_i. Then, we project the point from 2D to 3D through $project_{2D \to 3D}$ to obtain 3D point (x_i, y_i, z_i) in the 3D point cloud $\hat{y}_{cons}$:

$$x_i = \frac{d_i^*(u_i - o_x)}{f}, y_i = \frac{d_i^*(v_i - o_y)}{f}, z_i = d_i^*, \tag{15}$$

where f is the focal length, o_x and o_y are the 2D position of camera center, $d_i^* = d_i + \varepsilon$, ε is a shift to convert relative depth value d_i to the absolute depth value from the camera center. Please note that, a single image has infinite possible 3D reconstruction depending on different camera parameters. Since our objective of the 3D completion model is to learn the structure and the depth relationship in the 3D space, we do not need to restore exactly the same setting as the image being captured in the real world. On the other hand, as we cannot know the camera parameters of the real data, we hence set up reasonable projection parameters on our own and use the same setting in training the 3D completion model and finding 3D-aware pseudo-labels. In experiments, we adopt the same focal length f as virtual KITTI [13] and set ε as 40. Normally, ε is set equal to the focal length f, but such setting would lead to large values for x_i and y_i coordinates as indicated in Eq. (15). We therefore in experiments adopt the normalized depth values and fix the focal length to seek for a suitable shift ε, which gives a reasonable scale of 3D coordinates and still maintains the relationship between depth values.

Projection from 3D to 2D. After the 3D completion process, we obtain $\tilde{p}_{dense} = (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$, and then we project each point back to the original 2D plane as $(\tilde{u}_i, \tilde{v}_i)$ with the updated depth value $\tilde{d}_i = \tilde{z}_i - \varepsilon$ (we ignore the ε for simplicity in the main paper) by the inverse operation of Eq. (15):

$$\tilde{u}_i = \frac{\tilde{x}_i \cdot f}{\tilde{z}_i} + o_x, \tilde{v}_i = \frac{\tilde{y}_i \cdot f}{\tilde{z}_i} + o_y, \tilde{d}_i = \tilde{z}_i - \varepsilon, \tag{16}$$

where $(\tilde{u}_i, \tilde{v}_i)$ are rounded to integers. Since all the 3D points are generated through the completion model, the position $(\tilde{u}_i, \tilde{v}_i)$ projected from point cloud may be duplicated (i.e., two projected points happen to overlap in the 2D plane) or out of the original image plane (i.e., $(\tilde{u}_i, \tilde{v}_i) < 0$ or $(\tilde{u}_i, \tilde{v}_i) > (H, W)$). For duplicated points, we choose the minimum depth value among all duplicated points as the final depth. For those positions out of the original image plane, we view them as failing projection and do not take them as the pseudo-label $\hat{y}_{comp}$.

9.2   Discussions of 2D/3D Pseudo-label

We discuss whether $\hat{y}_{comp}$ is complementary to the original pseudo label $\hat{y}_{cons}$. We observe that in Figure 3 of the main paper, for some regions that look similar, the values of the same pixel between $\hat{y}_{cons}$ and $\hat{y}_{comp}$ are very close but have a little scale shift ($< 10^{-1}$). For the areas that appear different (e.g., bottom-left area), $\hat{y}_{comp}$ has nearer depth values than original $\hat{y}_{cons}$, in which nearer depth values are more reasonable for the object and grass in the bottom-left corner of the image. It shows that the completion model refers to the 3D structural information to produce better results.

10   Limitations

Figure 7 shows one example of the limitation in our 3D-PL with the stereo-pair setting. Since 3D-PL focuses on the structural information, it can perform well

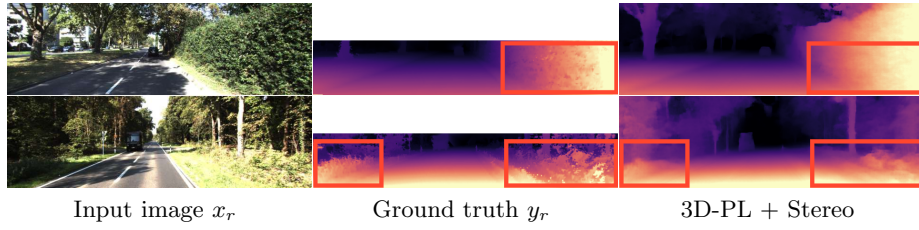Input image x_r Ground truth y_r 3D-PL + Stereo

Fig. 7: 3D-PL produces better results in overall structure and shape of objects, but may lose some details for the objects with complicated textures such as grass and plants.
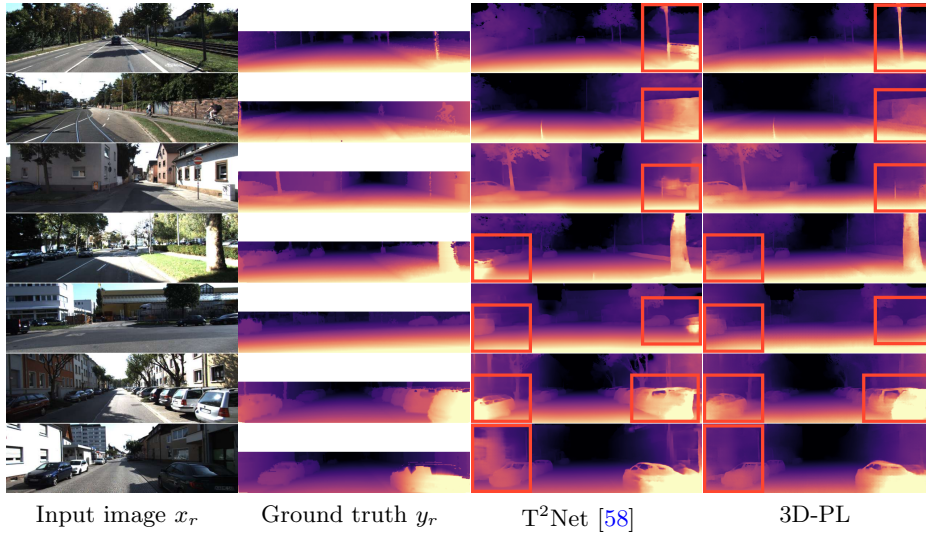

Input image x_r Ground truth y_r T^{2}Net [58]   3D-PL

Fig. 8: More qualitative results on KITTI [15] in the single-image setting.

on the overall structure, e.g., the shape of cars and the hard objects such as road signs or traffic lights. However, for the object that has complicated textures like grass, 3D-PL produces smoother results but loses the details of the plant.

11   More qualitative results

We provide more qualitative results for different settings. Figure 8 and Figure 9 are results for KITTI [15] in the single-image and stereo-pair settings, respectively. Figure 10 and Figure 11 are results for KITTI stereo 2015 [34] in single-image and stereo-pair settings, respectively. Figure 12 presents results for make3D [43] in the single-image setting.
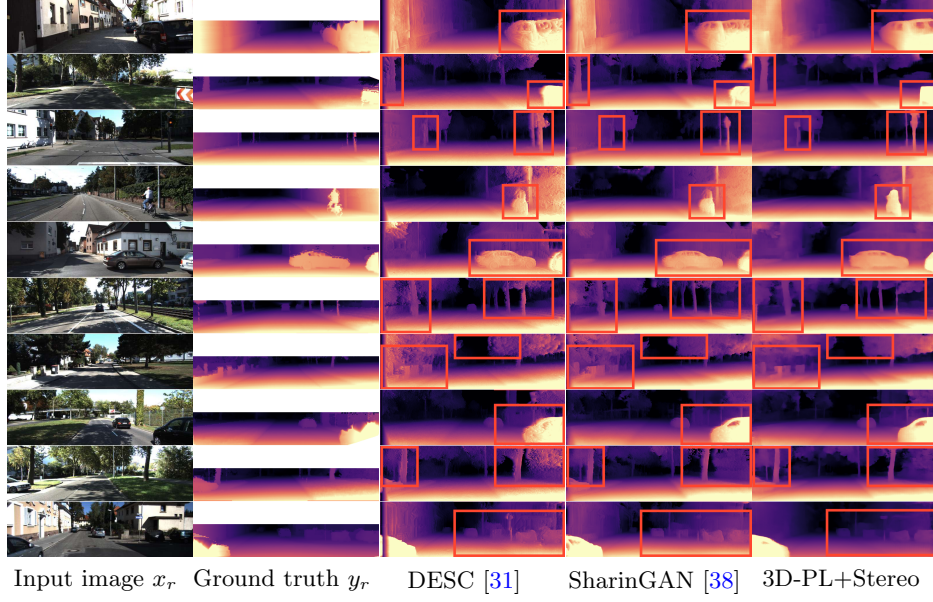
Input image x_r Ground truth y_r DESC [31]   SharinGAN [38]   3D-PL+Stereo

Fig. 9: More qualitative results on KITTI [15] with having stereo pairs during training.

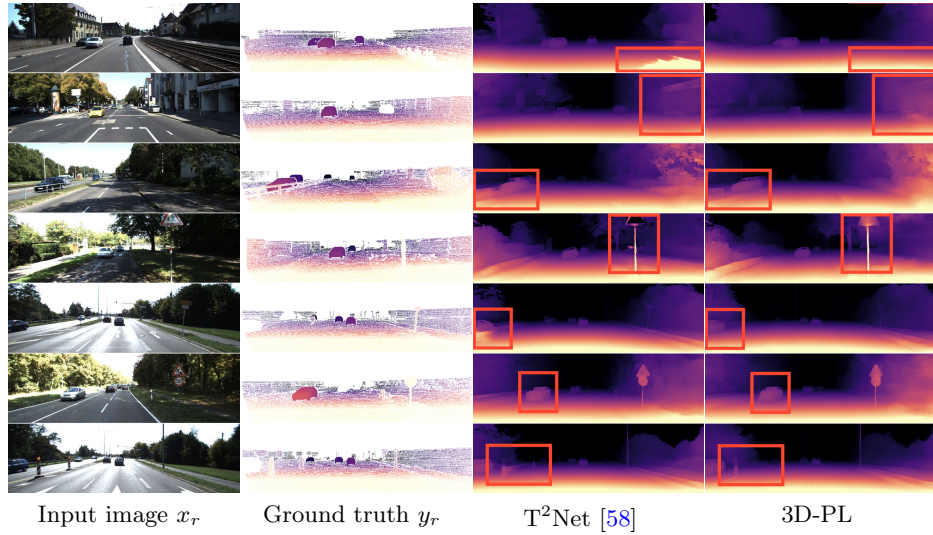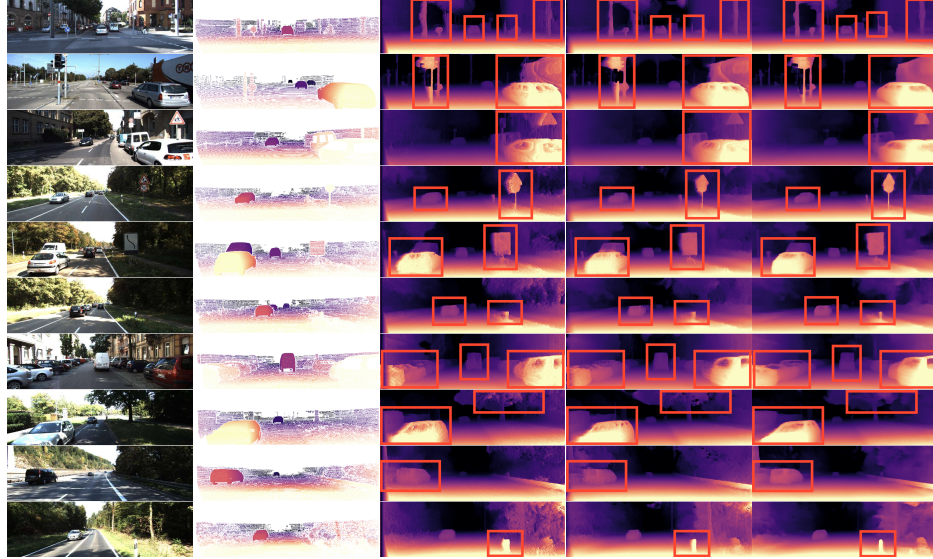Input image x_r Ground truth y_r T^{2}Net [58]   3D-PL
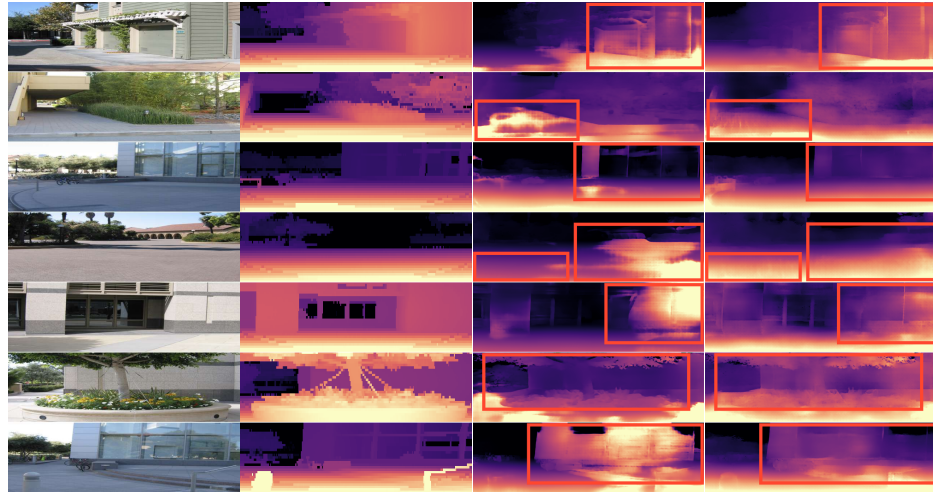
Fig. 10: More qualitative results on KITTI stereo 2015 [34] in the single-image setting.

Input image x_r Ground truth y_r DESC [31]   SharinGAN [38]   3D-PL+Stereo

Fig. 11: More qualitative results on KITTI stereo 2015 [34] with having stereo pairs during training.

Input image x_r Ground truth y_r T^{2}Net [58]   3D-PL

Fig. 12: More qualitative results on Make3D [43] in the single-image setting.