

# Mining Supervision for Dynamic Regions in Self-Supervised Monocular Depth Estimation

Hoang Chuong Nguyen<sup>1</sup> Tianyu Wang<sup>1</sup> Jose M. Alvarez<sup>2</sup> Miaomiao Liu<sup>1</sup>  
<sup>1</sup>Australian National University <sup>2</sup>NVIDIA

{hoangchuong.nguyen, tianyu.wang2, miaomiao.liu}@anu.edu.au josea@nvidia.com

## Abstract

This paper focuses on self-supervised monocular depth estimation in dynamic scenes trained on monocular videos. Existing methods jointly estimate pixel-wise depth and motion, relying mainly on an image reconstruction loss. Dynamic regions<sup>1</sup> remain a critical challenge for these methods due to the inherent ambiguity in depth and motion estimation, resulting in inaccurate depth estimation. This paper proposes a self-supervised training framework exploiting pseudo depth labels for dynamic regions from training data. The key contribution of our framework is to decouple depth estimation for static and dynamic regions of images in the training data. We start with an unsupervised depth estimation approach, which provides reliable depth estimates for static regions and motion cues for dynamic regions and allows us to extract moving object information at the instance level. In the next stage, we use an object network to estimate the depth of those moving objects assuming rigid motions. Then, we propose a new scale alignment module to address the scale ambiguity between estimated depths for static and dynamic regions. We can then use the depth labels generated to train an end-to-end depth estimation network and improve its performance. Extensive experiments on the Cityscapes and KITTI datasets show that our self-training strategy consistently outperforms existing self-/unsupervised depth estimation methods. Our code is available at <https://github.com/HoangChuongNguyen/mono-consistent-depth.git>

## 1. Introduction

Estimating a dense depth map from a monocular image is crucial for various applications, such as 3D reconstruction [23], autonomous driving [24], and augmented reality [30]. Self-/unsupervised depth estimation using monocular video sequences is an effective way to bypass the need for expensive depth labels [3, 8, 10]. Current methods minimize the image reconstruction loss during training to jointly estimate depth and camera pose [8, 10, 12, 13]. These meth-

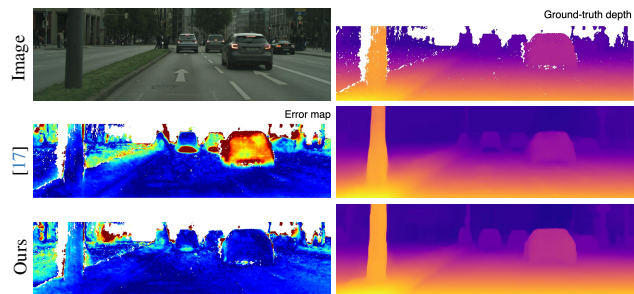


Figure 1. Depth predictions for an image. Areas with more intense red color in the error maps represent higher error. Our method produces more accurate depths for moving objects in the image.

ods suffer accuracy degradation in the presence of moving objects, e.g. cars, as the reconstruction process does not take motion into consideration. To address this limitation, a few works infer the camera and pixel-wise object motion in 3D, and the depth to reduce the image reconstruction error [15, 19]. Other approaches incorporate binary masks for moving objects to improve the pixel-wise motion estimation [3, 17]. Despite the accuracy improvement, as we will show in Section 3, there are inherent ambiguities entangling motion and depth estimation in dynamic regions that can not be fully resolved using regularization terms or joint optimization, i.e., enforcing sparsity on pixel-wise motion. In contrast, we propose to exploit *scale-consistent* pseudo depth labels for moving objects which we can then use to train a monocular depth estimation network.

In contrast to existing methods, we propose to decouple the depth estimation for static and dynamic regions in images. We first leverage existing self-supervised depth estimation frameworks to jointly estimate the depth, pixel-wise object motion, and camera rigid motion [19]. We use pixel-wise object motion to identify moving objects, e.g., the foreground, and their pixel-wise correspondences across frames. For the rest of the image, the background, we directly use the inferred depth map as depth labels.

For the foreground, the goal is to obtain pseudo depth la-

<sup>1</sup>Dynamic regions indicate regions covered by moving objects.

bels for each object that are scale-consistent with the background. We achieve this in two steps. First, we employ an Object Network to estimate the depth label for the dynamic regions for each frame in the sequence. This network takes a pair of images with corresponding objects as input to jointly estimate the relative camera motion, namely, rigid rotation and translation, and depth map for the object in the reference frame via the image reconstruction loss. This can be viewed as a dense structure-from-motion process that leads to scaled depth and camera pose estimation. Second, we propose a scale alignment network to resolve scale ambiguities between each object in the foreground and the background. We then obtain scale-consistent depth pseudo labels for the foreground that can be used as additional supervision signals for training depth estimation networks.

We demonstrate the benefits of using depth pseudo labels obtained using our approach on several existing Self-/unsupervised monocular depth estimation frameworks on the KITTI and Cityscapes datasets. Our results show consistent overall improvements over frameworks and datasets with significant error reduction on dynamic regions. In summary, our contributions are listed below.

- We introduce a new scale-alignment network to solve scale-ambiguities among objects and the background.
- To the best of our knowledge, we are the first to extract scale-consistent pseudo depth labels as supervision signal to solve the scale-ambiguity problem commonly present in monocular depth estimation for dynamic scenes.

Our approach consistently outperforms all previous self-/unsupervised depth estimation methods by large margins for dynamic regions. Compared to prior works, we achieve up to **52.6%** and **14.4%** error reduction on the Cityscapes and KITTI datasets, respectively.

## 2. Related Work

**Self-/unsupervised Learning from Stereo Images.** To avoid costly ground-truth data for training a depth network, [6] introduces an approach under the unsupervised framework for depth estimation using a synchronized stereo pair as training data. [7] imposes a consistency constraint on the predicted disparity when a stereo pair is adopted as a supervision signal. Generative adversarial networks are also explored for this task [1, 22]. [29] attempts to improve the training signal by aligning the lighting condition of the input images and modeling pixel photometric uncertainties. Recently, [26] proposed to estimate a depth distribution based on the concept of orthogonal planes in the 3D world. In this work, we show that we can exploit reliable depth supervision signals from monocular video instead of stereo pairs for practical applications.

**Self-/unsupervised Learning from Monocular Video.** The monocular video provides a less constrained form of self-/unsupervised learning. [32] firstly introduces a

promising framework that jointly learns depth and camera ego-motion from consecutive frames in a video. Later on, [8] improves the training signal by introducing the per-pixel minimum reprojection loss to filter out occluded pixels from the final image reconstruction loss. More robust geometric constraints enforcing consistency between predicted point clouds or predicted depth across consecutive frames are proposed by [2, 21]. To further boost the performance, several works propose new architectures for a depth estimation network [10, 12, 13, 31]. More recently, cost volume-based approaches that predict depth from multiple images are introduced [4, 11, 27]. Although significant improvements have been achieved, all the mentioned methods tend to have inferior performance in dynamic regions. The main challenge arises from the presence of moving objects in the scene, which does not satisfy the photometric consistency assumption given no specific consideration for object motion.

To address the issues caused by moving objects, [3] proposed a joint modelling of depth, camera motion, and object motion. Specifically, a warped image is first obtained by relying on the initially predicted depth and camera motion. This warped image and its corresponding reference image are then used to predict object rigid motion. [17] improves this framework by applying forward warping to encourage consistent geometry while modelling object motion. One disadvantage of such methods stems from the ambiguity of jointly predicting object depth and object motion. In particular, incorrect initial depth prediction could lead to incorrect warped images, which in turn causes errors in the predicted object motion. Nonetheless, there exist multiple combinations of depth and object motion that result in the minimization of the image reconstruction loss, namely the inherent ambiguities due to insufficient constraints [3, 17]. In lieu of predicting object rigid motion, [19] proposes to predict pixel-wise motion. Inspired by previous work, [15, 18] introduces a two-stage motion disentanglement, which is similar to [3] but with pixel-wise motion prediction. Nevertheless, these approaches exacerbate the scale ambiguity issue as there might be variations in the scale factor associated with different pixels of the same moving object. In this work, we propose to estimate high-quality scale-consistent depth pseudo labels for the moving objects and static background in separate stages which are used further to improve depth predictions for dynamic regions.

## 3. Preliminary

**Scale-consistent Depth Estimation.** Scale ambiguity is a phenomenon often encountered in self-/unsupervised dynamic scene monocular depth estimation. This setup involves two consecutive RGB frames, termed as reference and source frame  $\mathbf{I}_r, \mathbf{I}_s \in \mathbb{R}^{H \times W \times 3}$ , from a monocular video with known camera intrinsic matrix  $\mathbf{K}$  but unknown poses. For a static scene, the correspondence between pix-

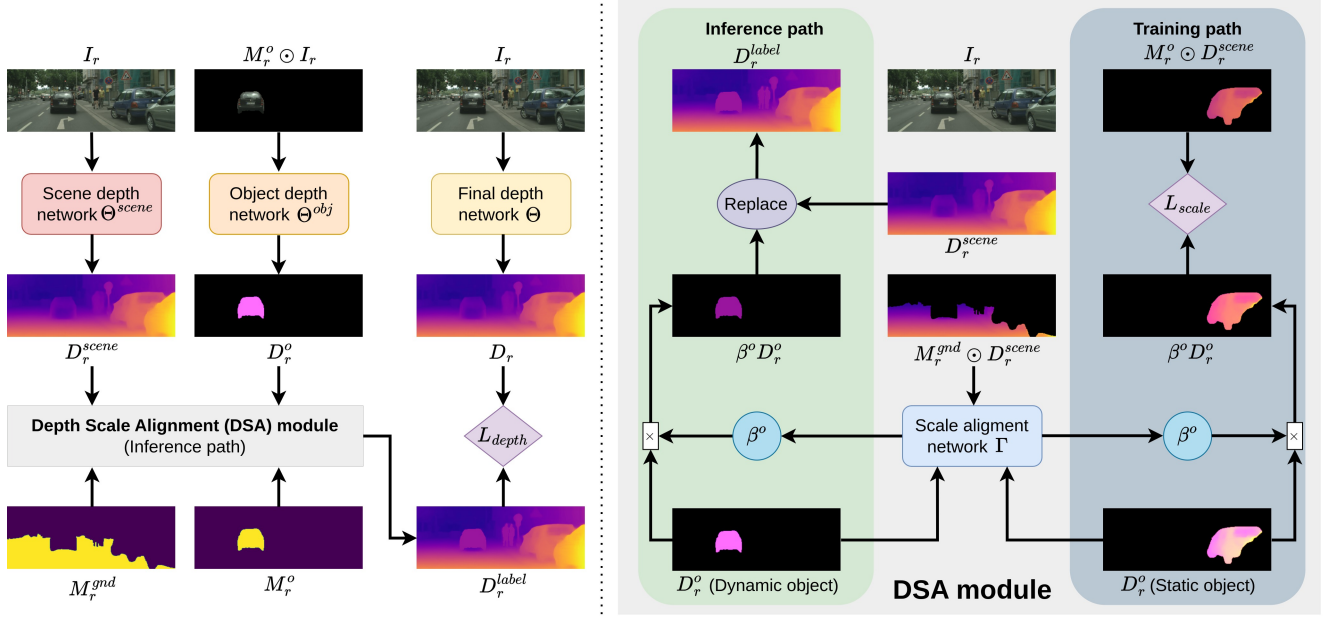


Figure 2. Our proposed framework. **Left:** How the pseudo depth label  $\mathbf{D}_r^{label}$  is extracted and used to train the final depth network. **Right:** How the DSA module is trained (training path) and utilized to produce  $\mathbf{D}_r^{label}$  (inference path). The DSA is trained to predict a scale  $\beta^o$  that minimizes the difference between two depth predictions of the same static object. It is then used to align the depth scale of every dynamic object to the depth scale of the static region.  $\mathbf{D}_r^{label}$  is constructed from depth of static regions in  $\mathbf{D}_r^{scene}$  and depth of dynamic objects after alignment  $\beta^o \mathbf{D}_r^o$ .

els in these frames can be established via

$$\mathbf{x}'d' = \mathbf{K}(\mathbf{R}\mathbf{K}^{-1}\mathbf{x}d + \mathbf{T}), \quad (1)$$

where  $\mathbf{x}$  and  $\mathbf{x}'$  is the homogeneous coordinate of a pixel in  $\mathbf{I}_r$  and  $\mathbf{I}_s$  respectively.  $d$  and  $d'$  represent the depth values for  $\mathbf{x}$  and  $\mathbf{x}'$  in their respective frames.  $\mathbf{R}$ ,  $\mathbf{T}$  specify (inferred) relative camera pose that is shared across all pixels. However, this mapping is not unique. One can scale both sides of the equation with a constant  $c$  for **all pixels** at the same time without violating the equality. That is,

$$\mathbf{x}'(cd') = \mathbf{K}(\mathbf{R}\mathbf{K}^{-1}\mathbf{x}(cd) + c\mathbf{T}). \quad (2)$$

Thus, the estimated depth  $d$  for all pixels is at best one unified scale away from the ground truth as the scale  $c$  can not be uniquely determined even with the correct correspondence. We say such depth prediction with a unified scale shared across all pixels is *scale-consistent*.

However, the presence of moving objects violates such scale consistency. To capture object motions, we need to introduce a translation vector  $\mathbf{t}_x$  that can vary from pixel to pixel and establish the correspondence as

$$\mathbf{x}'d' = \mathbf{K}(\mathbf{R}\mathbf{K}^{-1}\mathbf{x}d + \mathbf{T}) + \mathbf{t}_x \quad (3)$$

In this case, one can scale the depth of the pixel via an arbitrary constant  $c$  and always find a  $\mathbf{t}_x$  to retain the correspondence via

$$\mathbf{t}_x = d'\mathbf{x}' - \mathbf{K}(\mathbf{R}\mathbf{K}^{-1}\mathbf{x}(cd) + \mathbf{T}). \quad (4)$$

Thus, two pixels undergoing different motions can have different depth scales, which will lead to depth values with arbitrary differences. We refer to this phenomenon as *scale ambiguity* and term a depth prediction without unified scale as *scale-inconsistent*.

## 4. Method

Our goal is to estimate the depth map  $\mathbf{D}_r$  from a monocular image  $\mathbf{I}_r$ , namely  $\mathbf{D}_r = \Theta(\mathbf{I}_r)$  where  $\Theta$  denotes the depth estimation network. In particular, we aim to generate high-quality *scale-consistent* pseudo depth labels  $\mathbf{D}_r^{label}$  from a monocular video for self-supervised learning of  $\Theta$ . To this end, we propose to estimate a decoupled representation

$$\mathbf{D}_r^{label} = (1 - \mathbf{M}_r) \odot \mathbf{D}_r^{scene} + \mathbf{M}_r \odot \mathbf{D}_r^{dyn}, \quad (5)$$

where  $\mathbf{D}_r^{scene}$ , and  $\mathbf{D}_r^{dyn}$ , and  $\mathbf{M}_r$  represent depth label for the background scene, the moving objects, and moving object masks, respectively. An overview of our framework is shown in Fig. 2. In the following, we introduce the modules used to generate masks and depth labels employed above.

### 4.1. Scene Depth Estimation for $\mathbf{D}_r^{scene}$

In this stage, we aim to obtain the scene depth  $\mathbf{D}_r^{scene}$  for a reference image  $\mathbf{I}_r$  through unsupervised learning. At training time, given  $\mathbf{I}_r, \mathbf{I}_s \in \mathbb{R}^{H \times W \times 3}$ , from a monocular video with known  $\mathbf{K}$ , we formulate the problem as the minimization of image reconstruction error using depth ( $\mathbf{D}_r^{scene}$ ) and

camera pose  $(\mathbf{R}^{cam}, \mathbf{T}^{cam})$  as intermediate variables. To handle moving objects, following [19], we further introduce the pixel-wise object motion  $\Delta_r$  as an additional intermediate variable to generate a synthesized image  $\mathbf{I}_{r \leftarrow s}$ .

Instead of predicting these intermediate variables in parallel, we first adopt a depth network  $\Theta^{scene}$  and a camera pose network  $\Phi^{cam}$ , following the pipeline design in [17, 19], to predicate  $\mathbf{D}_r^{scene} \in \mathbb{R}^{H \times W \times 1}$  and  $\mathbf{R}^{cam}, \mathbf{T}^{cam}$  by  $\mathbf{D}_r^{scene} = \Theta^{scene}(\mathbf{I}_r)$ ;  $\mathbf{R}^{cam}, \mathbf{T}^{cam} = \Phi^{cam}(\mathbf{I}_r, \mathbf{I}_s)$ . Given the camera relative pose, a pixel  $\mathbf{x}$  in the reference frame with depth  $d = \mathbf{D}_r^{scene}(\mathbf{x})$  is mapped to  $\mathbf{x}'$  with depth  $d'$  in the source frame via the bijection  $\tau$  defined as

$$\mathbf{x}'d' = \tau(\mathbf{x}d) := \mathbf{K}(\mathbf{R}^{cam}\mathbf{K}^{-1}\mathbf{x}d + \mathbf{T}^{cam}). \quad (6)$$

Given  $\tau$ , we can compute a reconstruction of the source view  $\mathbf{I}_{r \rightarrow s}$  from the reference view  $\mathbf{I}_r$  via  $\mathbf{I}_{r \rightarrow s} = \mathcal{FW}(\tau, \mathbf{I}_r)$ , where  $\mathcal{FW}$  denotes the forward warping operation. We then employ a network  $\Psi$  to predict a motion map  $\Delta_r \in \mathbb{R}^{H \times W \times 3}$  taking  $\hat{\mathbf{I}}_{r \rightarrow s}$  and  $\mathbf{I}_s$  as input,

$$\Delta_r = \mathcal{FW}(\tau^{-1}, \Psi(\mathbf{I}_{r \rightarrow s}, \mathbf{I}_s)). \quad (7)$$

We get a refined correspondence  $\tau'$  defined as

$$\mathbf{x}''d'' = \tau'(\mathbf{x}d) := \tau(\mathbf{x}d) + \Delta_r(\mathbf{x}). \quad (8)$$

$\tau'$  induces a reference image reconstruction  $\mathbf{I}_{r \leftarrow s}$  via

$$\mathbf{I}_{r \leftarrow s} = \mathcal{BW}(\tau', \mathbf{I}_s), \quad (9)$$

where  $\mathcal{BW}$  denotes the backward warping operation.

The loss function for training the networks above includes the reference image reconstruction loss  $L_p$  [8], depth edge-aware smoothness loss  $L_s$  [7], and object-motion sparsity loss  $L_g$  [19]:

$$L_{scene} = L_p(\mathbf{I}_{r \leftarrow s}, \mathbf{I}_r) + L_s(\mathbf{D}_r^{scene}, \mathbf{I}_r) + L_g(\Delta_r), \quad (10)$$

where  $L_g(\cdot)$  encourages the predicted pixel-wise motion to be zero at static background regions in the scene. As discussed in Sec. 3, the static regions of  $\mathbf{D}_r^{scene}$  are scale-consistent.

## 4.2. Dynamic Object Depth Estimation for $\mathbf{D}_r^{label}$

At this stage, we focus on the scale-consistent depth estimation for moving object regions. As discussed in the previous section, we can obtain a pixel-wise motion map that leads to the estimation of the object masks (see Sec. 4.3 for details). Given the objects' masks and images of the object region only, we first estimate an initial object depth via unsupervised learning by minimizing the image reconstruction loss. Note that the estimated depth for moving objects is not scale consistent with the static regions. We therefore introduce a scale-alignment network to resolve the scale ambiguities. Below we provide details for these two steps.

**Self-supervised Object Depth Estimation.** Similar to the previous section, we treat depth and camera pose as the intermediate variables for unsupervised learning. Given the mask for object  $o$  in the reference view as  $\mathbf{M}_r^o \in \{0, 1\}^{H \times W \times 1}$ , we can map it to the source view and obtain the corresponding object mask as  $\mathbf{M}_{r \rightarrow s}^o = \mathcal{FW}(\tau', \mathbf{M}_r^o)$ .

We then estimate the depth and motion of the object  $o$  as

$$\mathbf{D}_r^o = \Theta^{obj}(\mathbf{M}_r^o \odot \mathbf{I}_r), \quad (11)$$

$$\mathbf{R}^o, \mathbf{T}^o = \Phi^{obj}(\mathbf{M}_r^o \odot \mathbf{I}_r, \mathbf{M}_{r \rightarrow s}^o \odot \mathbf{I}_s). \quad (12)$$

$\mathbf{R}^o$  and  $\mathbf{T}^o$  represent view changes caused by both object rigid motion and camera motion,  $\odot$  defines the element-wise multiplication operation. By substituting  $\mathbf{D}_r^o$ ,  $\mathbf{R}^o$ , and  $\mathbf{T}^o$  into Eq. 6, we can solve for pixel correspondence  $\tau^o$  and synthesize the masked reference image  $\mathbf{I}_{r \leftarrow s}^o$  via

$$\mathbf{I}_{r \leftarrow s}^o = \mathcal{BW}(\tau^o, \mathbf{M}_{r \rightarrow s}^o \odot \mathbf{I}_s). \quad (13)$$

The loss function used to train  $\Theta^{obj}$  and  $\Phi^{obj}$  is,

$$L_{obj} = L_p(\mathbf{I}_{r \leftarrow s}^o, \mathbf{M}_r^o \odot \mathbf{I}_r) + L_s(\mathbf{M}_r^o \odot \mathbf{D}_r^o, \mathbf{M}_r^o). \quad (14)$$

**Depth Scale Alignment.** In the previous stage, we unified the depth scale within each object. However, the scales still mismatch between different moving objects and static regions in  $\mathbf{D}_r^{scene}$ . To align the scales, we introduce the Depth Scale Alignment (DSA) module. The right part of Fig. 2 depicts the training process and how this module is utilized to address the scale ambiguity issue for moving objects.

The DSA module is designed based on three key observations. **Obs. 1:** In  $\mathbf{D}_r^{scene}$ , depth values of pixels within static objects exhibit a unified scale while moving objects still exhibit pixel-level scale-inconsistency due to the pixel-wise motion. By contrast,  $\Theta^{obj}$  can predict depth values for all the pixels at a unified scale within each static as well as the moving object. Thus, for static object only, the depth values predicated by  $\Theta^{scene}$  and  $\Theta^{obj}$  only differ by one single scale whereas it does not hold for moving objects.

**Obs. 2:** The scale-consistent depths of objects should have the similar range as the those of static regions they are situated on or close to. Thus, the objects' depth scale ambiguity could be resolved based on the information from the depth value of its nearby static regions. **Obs. 3:** Moving objects are indistinguishable from static objects when only one frame is observed, as the motion information is lost. Thus, a single frame model trained to predict the scale ratio between the inferred depth from  $\Theta^{scene}$  and  $\Theta^{obj}$  for static object can be generalized to moving objects after training.

The above observations indicate that we are able to train a DSA network  $\Gamma$  to predict a scale alignment ratio  $\beta \in \mathbb{R}^+$  to align the depth of a static object inferred by  $\Theta^{obj}$  to its desired depth predicted by  $\Theta^{scene}$ . In other words, ground

truth of the aligned depth scale for static objects are known. We thus only train the network  $\Gamma$  relying on static objects. Below we describe the training of  $\Gamma$ .

We denote the set of static objects as  $O^s$  and moving objects as  $O^d$  with  $O = O^s \cup O^d$  which are obtained based on the segmentation method described in 4.3. As stated in Obs. 1, we only consider **static** objects  $O^s$  in training. For each static object  $o \in O^s$ , as suggested by Obs. 2,  $\Gamma$  takes  $\mathbf{I}_r$ ,  $\mathbf{D}_r^o$ , and ground depth to infer scale alignment ratio  $\beta^o$ :

$$\beta^o = \Gamma(\mathbf{I}_r, \mathbf{D}_r^o, \mathbf{M}_r^{gnd} \odot \mathbf{D}_r^{scene}). \quad (15)$$

The loss function computes the difference between the depth predicted by  $\Theta^{obj}$  and  $\Theta^{scene}$  but after alignment as

$$L_{scale} = \frac{\sum_{u,v} |\beta^o \mathbf{D}_r^o(u,v) - \mathbf{M}_r^o(u,v) \mathbf{D}_r^{scene}(u,v)|}{\sum_{u,v} \mathbf{M}_r^o(u,v)}, \quad (16)$$

where the scale alignment is done by multiplying depth prediction  $\mathbf{D}_r^o(u,v)$  by  $\beta^o$ .

After training, Obs. 3 indicates that we can perform the inference of the DSA network on **moving** objects and obtain the depth ratio between each **moving** object and the static regions to achieve a dynamic region depth prediction  $\mathbf{D}_r^{dyn} = \sum_{o \in O^d} \beta^o \mathbf{M}_r^o \odot \mathbf{D}_r^o$  that share the same scale with the static regions within the same image. We define the dynamic region mask as  $\mathbf{M}_r = \sum_{o \in O^d} \mathbf{M}_r^o$ . Then, an image depth prediction as the depth label  $\mathbf{D}_r^{label}$  with a unified scale is generated based on Eq. 5. The depth scale alignment component is the key to achieving scale-consistent full image depth prediction and improves depth prediction accuracy by a large margin as demonstrated in the experiment section below.

### 4.3. Mask Generation $\mathbf{M}_r^o, \mathbf{M}_r^{gnd}$

The masks for objects and the ground in the image are key components to obtain the scale-consistent depth estimation. In principle, we can adopt existing instance segmentation methods, a pre-trained segmentation model such as *SEEM* [33] for objects and ground which leverages annotations, or build a self-supervised pipeline using existing unsupervised approaches to discover the object and ground masks. Below, we introduce self-supervised pipelines that can discover object masks and ground masks, respectively, to demonstrate that we can derive the segmentation masks in a self-supervised manner.

**Self-supervised Object Detection.** We adopt the slot-attention (SA) model [20] to obtain the pseudo moving object masks from the object motion map and depth map we obtained in Section 4.1. Given the extracted dynamic object masks, we could train a *MaskRCNN* using these object masks to discover all the object regions ( $\mathbf{M}_r^o$ ) including static ones which are required in the scale-alignment process. Details are provided in the Supplementary material.

**Self-supervised Ground Segmentation.** To obtain ground pseudo masks, we follow the method introduced by [28] and segment ground pixels according to their height in the 3D world and surface normal. Then, we train a segmentation model to predict the pseudo ground mask  $\mathbf{M}_r^{gnd}$  with the binary-cross entropy loss.

We show in Fig. 3 a qualitative comparison between the pseudo label, self-supervised segmentation, and *SEEM* segmentation results. Compared with the pseudo label, the self-supervised model is able to discover both static and dynamic moving objects with similar results using *SEEM*. Please see supplementary for more details.

### 4.4. Final Depth Prediction $\mathbf{D}_r$

Getting the scale-consistent depth prediction  $\mathbf{D}_r^{label}$  involves the evaluation of multiple networks. We propose to train a depth network  $\Theta$  with  $\mathbf{D}_r^{label}$  to directly predict scale-consistent depth from an RGB image in one go,

$$\mathbf{D}_r = \Theta(\mathbf{I}_r). \quad (17)$$

Although the loss function used to train this model can be as simple as a regression loss between its predictions and the pseudo label  $\mathbf{D}_r^{label}$ , we find that loss terms employed in the first stage prevent the model from learning the inaccurate pseudo depth label (e.g., induced by inaccurate object masks). The loss used to train the models is

$$L_{final} = L_p(\mathbf{I}_{r \leftarrow s}, \mathbf{I}_r) + L_s(\mathbf{D}_r, \mathbf{I}_r) + L_g(\Delta_r) + L_{depth}(\mathbf{D}_r, \mathbf{D}_r^{label}, \mathbf{M}_r). \quad (18)$$

The first three terms in Eq. 18 are identical to  $L_{scene}$  with  $L_{depth}$  provides scale-consistent dynamic region supervision and is defined as

$$L_{depth} = \frac{\sum_{u,v} \mathbf{M}_r(u,v) |\mathbf{D}_r(u,v) - \alpha \mathbf{D}_r^{label}(u,v)|}{\sum_{u,v} \mathbf{M}_r(u,v)}, \quad (19)$$

$$\alpha = \text{Median} \left( \left\{ \frac{\mathbf{D}_r(u,v)}{\mathbf{D}_r^{label}(u,v)} \right\}_{u,v} \right), \quad (20)$$

where  $\alpha$  is the median of the per-pixel depth ratio between  $\mathbf{D}_r$  and  $\mathbf{D}_r^{label}$  used to align the scale of  $\mathbf{D}_r^{label}$  to  $\mathbf{D}_r$ , effectively making  $L_{depth}$  scale-invariant. The rationale behind the design is that  $\mathbf{D}_r^{label}$ , as a scale-consistent pseudo label, is itself one scale away from the ground truth depth. A direct regression without  $\alpha$  will force our network to predict depth at the specific scale. However, as the scale itself is arbitrary, there is no reason to prefer one scale over the other. By applying  $\alpha$  we allow the network to find its own scale without introducing scale-inconsistency, lightening the learning burden.

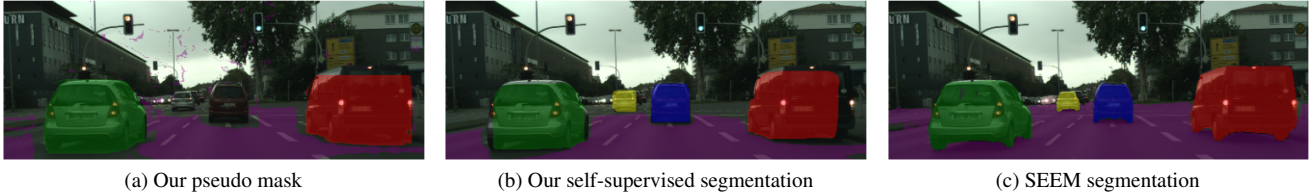


Figure 3. Visualization of segmentation over the same image. Static cars (two in the middle, stopped by the traffic light) ignored by the pseudo label are segmented out by both our self-supervised model as well as SEEM.

We initialize  $\Theta$  with  $\Theta^{scene}$ , i.e. instead of training from scratch, we fine-tune  $\Theta^{scene}$ . The camera pose network  $\Phi^{cam}$  and the pixel-wise motion network  $\Psi$  are also fine-tuned at this stage.

## 5. Experiments

### 5.1. Implementation Details

**Dataset and Evaluation Metrics** We evaluate our method on the Cityscapes and KITTI datasets. For Cityscapes, we use 69,731 images for training and 1,525 images for testing. For KITTI, we use 39,810 training images and evaluate our method on 697 images from the Eigen split [5]. Additionally, we manually annotate moving-object masks for test images for both datasets. The annotation allows us to evaluate performance on static and dynamic regions separately.

Tab. 1 reports the statistics on the number of dynamic objects. The table reveals that the KITTI dataset primarily consists of static scenes, making it less prone to scale ambiguity. On the other hand, the Cityscapes dataset features a large amount of moving objects. For this reason, our experiments mainly focus on the Cityscapes dataset. Regarding evaluation metrics, we adopt the absolute relative error (Abs Rel), square relative error (Sq Rel) and the accuracy metric for depth prediction ( $\delta < 1.25$ ) as used in [8].

Dataset	Dynamic objects/image	% Dynamic pixels
KITTI	0.828	0.6%
Cityscapes	2.266	2.9%

Table 1. Average number of dynamic objects and percentage of dynamic pixels in the Cityscapes and KITTI dataset.

**Training and Network Details** We perform the same data augmentation strategy as that is in [27]. For our camera and object pose networks, we adopt the network structure from [8]. We use RAFT [25] as our pixel-wise motion network. The DSA modules consist of a *ResNet18* [14] backbone followed by 3 fully connected (FC) layers. *ReLU* functions are applied after each FC layer except for the last one. We explore several depth network architectures in our experiments. For each stage, we train models for 30 epochs with Adam optimizer [16] with the learning rate set to 0.0002, halved every 15 epochs. Exact forms of the loss function  $L_p, L_s, L_g$  can be found in the supplementary material.

## 5.2. Results

**Performance Improvements:** Our method improves the performance of various self-/unsupervised monocular depth estimation networks, especially in dynamic regions, on the Cityscapes dataset. Specifically, we apply our method to a range of self-supervised monocular depth estimation network backbones, starting from the simple *ResNet18* (with randomized layer normalization [9]) to the more recent ones including *PackNet* [10], *DiffNet* [31] and *BrNet* [12]. The performances of each network architecture under different settings are reported in Tab. 2.

The quantitative results indicate that when the models are trained using photometric loss only, errors in dynamic regions are considerably large (the first rows of each backbone). Integrating pixel-wise object motion prediction improves the performance; however, the depth predictions for dynamic regions are not scale-consistent due to the scale ambiguity (the second rows of each backbone). Our pseudo depth label provides scale-consistent depth supervision, which further improves depth predictions across all backbones (the third row of each backbone).

The introduction of our models leads to an average of **52.1%** reduction in the square relative error for the dynamic region, and **2.8%** and **5.9%** reduction for static and all regions respectively. Note the performance of the network on the static regions largely reflects the performance of the network on all regions as the static regions occupy the majority area of the input image. Therefore, the error reduction on all-region is not as significant as that for dynamic regions only. We further show that by applying our method in conjunction with a pre-trained segmentation model, we reduce the square relative errors by **55.8%**, **4.0%**, **7.6%** on average in dynamic, static, and all regions, respectively (the last rows of each backbone).

**Baseline Comparison:** In Table 3, we compare our proposed framework against prior methods on the Cityscapes and KITTI dataset quantitatively. For works with no official pre-trained model release [8, 12, 19], we re-train the models with their official implementation and evaluate on static and dynamic regions separately. For [13, 15] we only report the all-region performance as there is no publicly available implementation or pre-trained models. For the rest works, we evaluate their pre-trained models directly. All models

Backbone	Pixel motion	Pseudo depth	Sematic prior	Dynamic region			Static region			All region		
				Abs Rel	Sq Rel	$\delta < 1.25$	Abs Rel	Sq Rel	$\delta < 1.25$	Abs Rel	Sq Rel	$\delta < 1.25$
Resnet18 [9]	✓ ✓ ✓	✓ ✓ ✓	✓	0.352	11.169	0.858	0.118	1.362	0.862	0.127	1.743	0.722
				0.144	1.650	0.898	0.095	0.986	0.900	0.097	0.982	0.854
				<u>0.107</u>	<u>0.784</u>	<b>0.906</b>	<u>0.093</u>	<u>0.961</u>	<b>0.907</b>	<u>0.094</u>	<u>0.931</u>	<u>0.907</u>
				<b>0.100</b>	<b>0.647</b>	<u>0.903</u>	<b>0.091</b>	<b>0.931</b>	<u>0.904</u>	<b>0.092</b>	<b>0.895</b>	<b>0.911</b>
Packnet [10]	✓ ✓ ✓	✓ ✓ ✓	✓	0.182	2.667	0.891	0.104	1.167	0.894	0.108	1.210	0.813
				0.132	0.987	<u>0.906</u>	0.091	0.861	<u>0.908</u>	0.093	0.844	0.857
				<u>0.112</u>	<u>0.703</u>	<b>0.908</b>	<u>0.090</u>	<b>0.858</b>	<b>0.909</b>	<u>0.091</u>	<u>0.831</u>	<u>0.890</u>
				<b>0.104</b>	<b>0.666</b>	<b>0.908</b>	<b>0.089</b>	<u>0.859</u>	<u>0.908</u>	<b>0.090</b>	<b>0.830</b>	<b>0.901</b>
DiffNet [31]	✓ ✓ ✓	✓ ✓ ✓	✓	0.233	4.369	0.893	0.103	1.109	0.896	0.107	1.187	0.771
				0.152	1.646	0.912	0.087	0.831	0.917	0.090	0.833	0.828
				<u>0.113</u>	<b>0.657</b>	<u>0.916</u>	<u>0.083</u>	<b>0.775</b>	<u>0.919</u>	<u>0.085</u>	<b>0.753</b>	<u>0.889</u>
				<b>0.105</b>	<u>0.692</u>	<b>0.919</b>	<b>0.082</b>	<u>0.781</u>	<b>0.921</b>	<b>0.083</b>	<u>0.757</u>	<b>0.893</b>
BrNet [12]	✓ ✓ ✓	✓ ✓ ✓	✓	0.224	3.440	0.883	0.105	1.052	0.889	0.106	1.124	0.707
				0.168	2.041	0.914	<u>0.082</u>	0.747	0.920	0.087	0.780	0.809
				<u>0.119</u>	<u>0.670</u>	<u>0.918</u>	<b>0.080</b>	<u>0.736</u>	<u>0.922</u>	<u>0.084</u>	<u>0.722</u>	<u>0.872</u>
				<b>0.106</b>	<b>0.577</b>	<b>0.921</b>	<b>0.080</b>	<b>0.717</b>	<b>0.923</b>	<b>0.081</b>	<b>0.696</b>	<b>0.892</b>

Table 2. Results of training different depth estimation networks backbone under different settings. For each backbone, the best results are in **bold**, the second best are underlined. **With pixel-wise:** The model is trained in our first stage only. **With pixel motion and pseudo depth:** Our fully self-supervised pipeline is applied to train the model. **With semantic prior:** Objects and ground masks are produced by a pre-trained segmentation model [33] instead of the self-supervised ones.

Method	Dataset	Sematic prior	Dynamic region			Static region			All region		
			Abs Rel	Sq Rel	$\delta < 1.25$	Abs Rel	Sq Rel	$\delta < 1.25$	Abs Rel	Sq Rel	$\delta < 1.25$
Monodepth2* [8]	C	✓	0.286	6.036	0.716	0.119	1.461	0.877	0.127	1.678	0.872
Li <i>et al.</i> * [19]			0.188	1.654	0.735	0.118	1.198	0.835	0.119	1.186	0.833
InstaDM [17]			0.189	2.538	0.795	0.102	1.058	0.895	0.106	1.104	0.890
RMDepth [15]			–	–	–	–	–	–	0.100	0.839	0.895
DaCNN [13]			–	–	–	–	–	–	0.113	1.380	0.888
ResNet18 [9] + Ours	C		<b>0.107</b>	<u>0.784</u>	<b>0.907</b>	<u>0.093</u>	<u>0.961</u>	<u>0.907</u>	<u>0.094</u>	<u>0.931</u>	<u>0.906</u>
PackNet [10] + Ours			<u>0.112</u>	<u>0.703</u>	<u>0.890</u>	<u>0.090</u>	<u>0.858</u>	<u>0.909</u>	<u>0.091</u>	<u>0.831</u>	<u>0.908</u>
DiffNet [31] + Ours			<u>0.113</u>	<b>0.657</b>	<u>0.889</u>	<u>0.083</u>	<u>0.775</u>	<u>0.919</u>	<u>0.085</u>	<u>0.753</u>	<u>0.916</u>
BrNet [12] + Ours			<u>0.119</u>	<u>0.670</u>	<u>0.872</u>	<b>0.080</b>	<b>0.736</b>	<b>0.922</b>	<b>0.084</b>	<b>0.722</b>	<b>0.918</b>
PackNet [10]	K		0.213	2.820	0.762	0.108	0.814	0.883	0.110	0.834	0.881
DiffNet [31]			0.177	2.072	0.930	0.101	0.743	<b>0.899</b>	0.102	0.753	0.897
BrNet* [12]			0.183	1.715	0.760	0.104	0.702	0.885	0.106	0.711	0.884
RMDepth [15]			–	–	–	–	–	–	0.108	0.710	0.884
DaCNN [13]			–	–	–	–	–	–	<b>0.099</b>	<b>0.661</b>	<b>0.897</b>
DiffNet [31] + Ours	K	✓	0.158	<u>1.468</u>	<u>0.838</u>	0.101	<u>0.687</u>	0.894	0.102	0.693	0.892
DiffNet [31] + Ours			<b>0.143</b>	<b>1.191</b>	<b>0.845</b>	<b>0.099</b>	<b>0.658</b>	0.895	0.100	0.662	0.894
BrNet [12] + Ours			<u>0.160</u>	<u>1.273</u>	<u>0.812</u>	0.103	<u>0.690</u>	<b>0.889</b>	0.103	0.692	0.888
BrNet [12] + Ours			<u>0.162</u>	<u>1.207</u>	<u>0.812</u>	0.102	<u>0.673</u>	<b>0.889</b>	0.103	0.675	0.888

Table 3. Comparison between the models trained with our method and previous works on Cityscapes (C) and Kitti (K) dataset. The best results are in **bold**. Additionally, the performance of our methods is underlined if it is better than all previous works. (\*) indicates results reproduced using the official code. All methods are trained on monocular videos.

are evaluated following the process detailed in [27].

The results show that, in the Cityscapes dataset, our models outperform all the previous methods on dynamic regions by significant margins across all metrics. For the Sq Rel metric, our models outperform the best baseline by a margin of **0.870**, that is, a **52.6%** error reduction. By resolving scale ambiguity under the presence of moving objects, our models achieve new state-of-the-art performance on the Cityscapes dataset. To be more specific, the model with *ResNet18* backbone achieves state-of-the-art on the Abs Rel and Sq Rel metrics. Our other models outperform

all previous works in all metrics across different types of regions. The improvement of the performance are even larger when a pre-trained segmentation model is available. Figure 4 shows the qualitative results on the Cityscapes dataset. Although [17] and [19] produce low-error predictions for static regions, they fail to predict accurate depth for the moving objects. In contrast, our methods are able to accurately estimate depth for both static and dynamic regions.

For KITTI dataset, our models outperform all previous works in dynamic regions and achieve a reduction of at least **14.4%** in the Sq Rel metric. Due to a small number of mov-

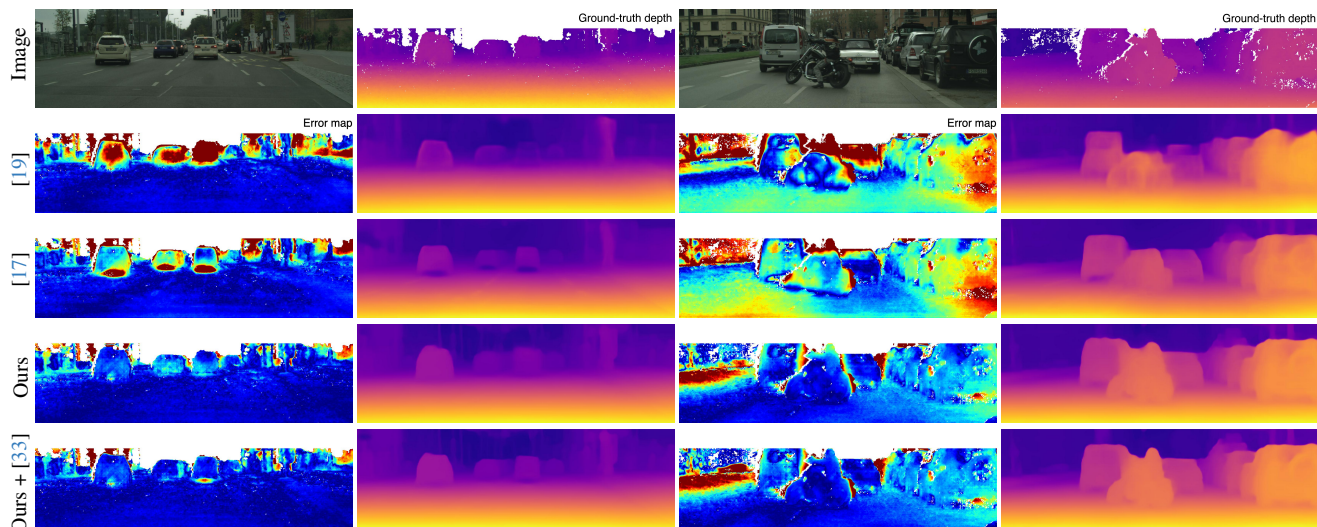


Figure 4. Qualitative results on Cityscapes. The error map is computed as the per-pixel relative absolute depth error. Red areas indicate high errors. While the depth maps are not clearly distinguishable, the error maps indicate that both [17] and [19] fail to predict accurate depth for moving objects in the images. In contrast, our models are able to predict the depth for these objects with smaller errors. Our model, in conjunction with a pre-trained segmentation model *SEEM* [33], can achieve even better predictions.

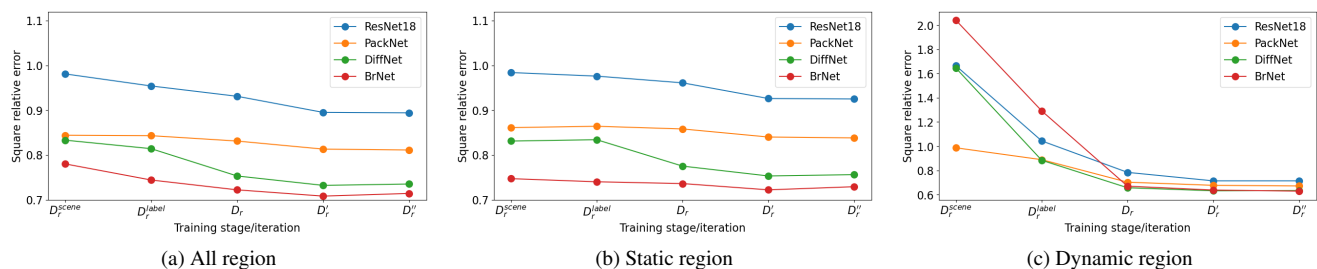


Figure 5. Depth prediction errors at different stage/self-training iteration. The largest performance gains are achieved while fine-tuning our method with the pseudo label  $D_r^{label}$ . Their performance slightly improves through our iterative training strategy.

ing objects in the image, scale-consistent depth predictions in dynamic regions result in marginal all-region improvements. While our models do not outperform *DaCNN*[13] in all-region<sup>2</sup>, we expect to achieve higher performance by combining our training method with their network design.

### 5.3. Iterative Self-training

In this section, we investigate, as a natural extension to our pipeline, a self-training setup where we use the depth prediction from  $\Theta$  as the pseudo depth label and finetune  $\Theta$  itself following the procedure described in Sec. 4.4. In Fig. 5, we visualize the change of errors of ours models at different stage/self-training iteration. In the figure, we use  $D_r'$  and  $D_r''$  to represent the depth prediction after one and two self-training iterations. The figure shows that the largest performance gain happens at fine-tuning with pseudo label  $D_r^{label}$ . The performance of our models slightly improve at the first iteration and remains stable after that.

<sup>2</sup>We cannot combine our training method with *DaCNN* [13] as is not publicly available.

## 6. Conclusion

In this work, we proposed a fully self-supervised pipeline that generates scale-consistent depth pseudo labels. Our decoupled depth estimation for static and dynamics regions can lead to a high-quality pseudo depth labels. Such labels have been used to improve the performance of a range of self-supervised monocular depth estimation frameworks, especially in regions with moving objects. Extensive experimental results demonstrate that we achieve the new state-of-the-art performance on the Cityscapes dataset, and consistently outperform previous works in depth predictions for dynamic regions on both KITTI and Cityscapes dataset. One limitation of our method is its generalizability to indoor scenes in which the main moving objects are humans whose motion cannot be described by a single rigid motion.

**Acknowledgement.** This research was supported in part by the Australia Research Council DECRA Fellowship (DE180100628) and ARC Discovery Grant (DP200102274).



## References

- [1] Filippo Aleotti, Fabio Tosi, Matteo Poggi, and Stefano Mattochia. Generative adversarial networks for unsupervised monocular depth prediction. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. [2](#)
- [2] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems*, 32, 2019. [2](#)
- [3] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8001–8008, 2019. [1](#), [2](#)
- [4] Xingyu Chen, Ruonan Zhang, Ji Jiang, Yan Wang, Ge Li, and Thomas H Li. Self-supervised monocular depth estimation: Solving the edge-fattening problem. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5776–5786, 2023. [2](#)
- [5] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. [6](#)
- [6] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII 14*, pages 740–756. Springer, 2016. [2](#)
- [7] Clément Godard, Oisín Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017. [2](#), [4](#)
- [8] Clément Godard, Oisín Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019. [1](#), [2](#), [4](#), [6](#), [7](#)
- [9] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8977–8986, 2019. [6](#), [7](#)
- [10] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2485–2494, 2020. [1](#), [2](#), [6](#), [7](#)
- [11] Vitor Guizilini, Rares Ambrus, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-frame self-supervised depth with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 160–170, 2022. [2](#)
- [12] Wencheng Han, Junbo Yin, Xiaogang Jin, Xiangdong Dai, and Jianbing Shen. Brnet: Exploring comprehensive features for monocular depth estimation. In *European Conference on Computer Vision*, pages 586–602. Springer, 2022. [1](#), [2](#), [6](#), [7](#)
- [13] Wencheng Han, Junbo Yin, and Jianbing Shen. Self-supervised monocular depth estimation by direction-aware cumulative convolution network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8613–8623, 2023. [1](#), [2](#), [6](#), [7](#), [8](#)
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [6](#)
- [15] Tak-Wai Hui. Rm-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1675–1684, 2022. [1](#), [2](#), [6](#), [7](#)
- [16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [6](#)
- [17] Seokju Lee, Sunghoon Im, Stephen Lin, and In So Kweon. Learning monocular depth in dynamic scenes via instance-aware projection consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1863–1872, 2021. [1](#), [2](#), [4](#), [7](#), [8](#)
- [18] Seokju Lee, Francois Rameau, Fei Pan, and In So Kweon. Attentive and contrastive learning for joint depth and motion field estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4862–4871, 2021. [2](#)
- [19] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. In *Conference on Robot Learning*, pages 1908–1917. PMLR, 2021. [1](#), [2](#), [4](#), [6](#), [7](#), [8](#)
- [20] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. [5](#)
- [21] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5667–5675, 2018. [2](#)
- [22] Andrea Pilzer, Dan Xu, Mihai Puscas, Elisa Ricci, and Nicu Sebe. Unsupervised adversarial depth estimation using cycled generative networks. In *2018 international conference on 3D vision (3DV)*, pages 587–595. IEEE, 2018. [2](#)
- [23] Kyle Sargent, Jing Yu Koh, Han Zhang, Huiwen Chang, Charles Herrmann, Pratul Srinivasan, Jiajun Wu, and Deqing Sun. Vq3d: Learning a 3d-aware generative model on imagenet. *arXiv preprint arXiv:2302.06833*, 2023. [1](#)
- [24] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. [1](#)

- [25] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 6
- [26] Ruoyu Wang, Zehao Yu, and Shenghua Gao. Planedepth: Self-supervised depth estimation via orthogonal planes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21425–21434, 2023. 2
- [27] Jamie Watson, Oisín Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1164–1174, 2021. 2, 6, 7
- [28] Feng Xue, Guirong Zhuo, Ziyuan Huang, Wufei Fu, Zhuoyue Wu, and Marcelo H Ang. Toward hierarchical self-supervised monocular absolute depth estimation for autonomous driving applications. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2330–2337. IEEE, 2020. 5
- [29] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1281–1292, 2020. 2
- [30] Zhoutong Zhang, Forrester Cole, Richard Tucker, William T Freeman, and Tali Dekel. Consistent depth of moving objects in video. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021. 1
- [31] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. *arXiv preprint arXiv:2110.09482*, 2021. 2, 6, 7
- [32] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 2
- [33] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *arXiv preprint arXiv:2304.06718*, 2023. 5, 7, 8