

# Towards Scale-Aware, Robust, and Generalizable Unsupervised Monocular Depth Estimation by Integrating IMU Motion Dynamics

Sen Zhang<sup>✉</sup>, Jing Zhang<sup>✉</sup>, and Dacheng Tao<sup>✉</sup>

The University of Sydney, Sydney, Australia  
szha2609@uni.sydney.edu.au  
{jing.zhang1, dacheng.tao}@sydney.edu.au

**Abstract.** Unsupervised monocular depth and ego-motion estimation has drawn extensive research attention in recent years. Although current methods have reached a high up-to-scale accuracy, they usually fail to learn the true scale metric due to the inherent scale ambiguity from training with monocular sequences. In this work, we tackle this problem and propose DynaDepth, a novel scale-aware framework that integrates information from vision and IMU motion dynamics. Specifically, we first propose an IMU photometric loss and a cross-sensor photometric consistency loss to provide dense supervision and absolute scales. To fully exploit the complementary information from both sensors, we further drive a differentiable camera-centric extended Kalman filter (EKF) to update the IMU preintegrated motions when observing visual measurements. In addition, the EKF formulation enables learning an ego-motion uncertainty measure, which is non-trivial for unsupervised methods. By leveraging IMU during training, DynaDepth not only learns an absolute scale, but also provides a better generalization ability and robustness against vision degradation such as illumination change and moving objects. We validate the effectiveness of DynaDepth by conducting extensive experiments and simulations on the KITTI and Make3D datasets. (code)

**Keywords:** Unsupervised Monocular Depth Estimation, Differentiable Camera-Centric EKF, Visual-Inertial SLAM, Ego-motion Uncertainty

## 1 Introduction

Monocular depth estimation is a fundamental computer vision task which plays an essential role in many real-world applications such as autonomous driving, robot navigation, and virtual reality [36, 20, 44]. Classical geometric methods resolve this problem by leveraging the geometric relationship between temporally contiguous frames and formulating depth prediction as an optimization problem [9, 29, 8]. While geometric methods have achieved good performance, they are sensitive to either textureless regions or illumination changes. The computational cost for dense depth prediction also limits their practical use. Recently deep learning techniques have reformed this research field by training networks to predict depth

directly from monocular images and designing proper losses based on ground-truth depth labels or geometric depth clues from visual data. While supervised learning methods achieve the best performance [7,24,11,1,45], the labour cost for collecting ground-truth labels prohibits their use in real-world. To address this issue, unsupervised monocular depth estimation has drawn a lot of research attention [48,14], which leverages the photometric error from backwarping.

Although unsupervised monocular depth learning has made great progress in recent years, there still exist several fundamental problems that may obstruct its usage in real-world. First, current methods suffer from the scale ambiguity problem since the backwarping process is equivalent up to an arbitrary scaling factor w.r.t. depth and translation. While current methods are usually evaluated by re-scaling each prediction map using the median ratio between the ground-truth depth and the prediction, it is difficult to obtain such median ratios in practice. Secondly, it is well-known that the photometric error is sensitive to illumination change and moving objects, which violate the underlying assumption of the backwarping projection. In addition, though uncertainty has been introduced for the photometric error map under the unsupervised learning framework [21,41], it remains non-trivial to learn an uncertainty measure for the predicted ego-motion, which could further benefit the development of a robust and trustworthy system.

In this work, we tackle the above-mentioned problems and propose DynaDepth, a novel scale-aware monocular depth and ego-motion prediction method that explicitly integrates IMU *motion dynamics* into the vision-based system under a camera-centric extended Kalman filter (EKF) framework. Modern sensor suites on vehicles that collect data for training neural networks usually contain multiple sensors beyond cameras. IMU presents a commonly-deployed one which is advantageous in that (1) it is robust to the scenarios when vision fails such as in illumination-changing and textureless regions, (2) the absolute scale metric can be recovered by inquiring the IMU motion dynamics, and (3) it does not suffer from the visual domain gap, leading to a better generalization ability across datasets. While integrating IMU information has dramatically improved the performance of classical geometric odometry and simultaneous localization and mapping (SLAM) systems [28,22,31], its potential in the regime of unsupervised monocular depth learning is much less explored, which is the focus of this work.

Specifically, we propose a scale-aware IMU photometric loss which is constructed by performing backwarping using ego-motion integrated from IMU measurements, which provides dense supervision by using the appearance-based photometric loss instead of naively constraining the ego-motion predicted by networks. To accelerate the training process, the IMU preintegration technique [26,10] is adopted to avoid redundant computation. To correct the errors that result from illumination change and moving objects, we further propose a cross-sensor photometric consistency loss between the synthesized target views using network-predicted and IMU-integrated ego-motions, respectively. Unlike classical visual-inertial SLAM systems that accumulate the gravity and the velocity estimates from initial frames, these two metrics are unknown for the image triplet used in unsupervised depth estimation methods. To address this issue, DynaDepth

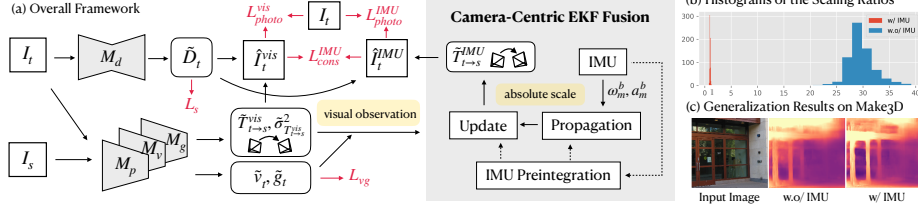


Fig. 1: (a) The overall framework of DynaDepth.  $\hat{I}_t^{vis}$  and  $\hat{I}_t^{IMU}$  denote the reconstructed target frames from the source frame  $I_s$ . Detailed notations of other terms are given in Section 3. (b) Histograms of the scaling ratios between the medians of depth predictions and the ground-truth. (c) Generalization results on Make3D using models trained on KITTI with (w/) and without (w.o/) IMU.

trains two extra lightweight networks that take two consecutive frames as input and predict the camera-centric gravity and velocity during training.

Considering that IMU and camera present two independent sensing modalities that complement each other, we further derive a differentiable camera-centric EKF framework for DynaDepth to fully exploit the potential of both sensors. When observing new ego-motion predictions from visual data, DynaDepth updates the preintegrated IMU terms based on the propagated IMU error states and the covariances of visual predictions. The benefit is two-fold. First, IMU is known to suffer from inherent noises, which could be corrected by the relatively accurate visual predictions. Second, fusing with IMU under the proposed EKF framework not only introduces scale-awareness, but also provides an elegant way to learn an uncertainty measure for the predicted ego-motion, which can be beneficial for recently emerging research methods that incorporate deep learning into classical SLAM systems to achieve the synergy of learning, geometry, and optimization.

Our overall framework is shown in Fig. 1. In summary, our contributions are:

- We propose an IMU photometric loss and a cross-sensor photometric consistency loss to provide dense supervision and absolute scales
- We derive a differentiable camera-centric EKF framework for sensor fusion.
- We show that DynaDepth benefits (1) the learning of the absolute scale, (2) the generalization ability, (3) the robustness against vision degradation such as illumination change and moving objects, and (4) the learning of an ego-motion uncertainty measure, which are also supported by our extensive experiments and simulations on the KITTI and Make3D datasets.

## 2 Related Work

### 2.1 Unsupervised Monocular Depth Estimation

Unsupervised monocular depth estimation has drawn extensive research attention recently [48, 27, 14], which uses the photometric loss by backwarping

adjacent images. Recent works improve the performance by introducing multiple tasks [42,32,19], designing more complex networks and losses [18,15,38,49], and constructing the photometric loss on learnt features [35]. However, monocular methods suffer from the scale ambiguity problem. DynaDepth tackles this problem by integrating IMU dynamics, which not only provides absolute scale, but also achieves state-of-the-art accuracy even if only lightweight networks are adopted.

## 2.2 Scale-Aware Depth Learning

Though supervised depth learning methods [7,11,1] can predict depths with absolute scale, the cost of collecting ground-truth data limits its practical use. To relieve the scale problem, local reprojected depth consistency loss has been proposed to ensure the scale consistency of the predictions [2,47,43]. However, the absolute scale is not guaranteed in these methods. Similar to DynaDepth, there exist methods that resort to other sensors than monocular camera, such as stereo camera that allows a scale-aware left-right consistency loss [13,14,46], and GPS that provides velocities to constrain the ego-motion network [15,3]. In comparison with these methods, using IMU is beneficial in that (1) IMU provides better generalizability since it does not suffer from the visual domain gap, and (2) unlike GPS that cannot be used indoors and cameras that fail in texture-less, dynamic and illumination changing scenes, IMU is more robust to the environments.

## 2.3 Visual-Inertial SLAM Systems

The fusion of vision and IMU has achieved great success in classical visual-inertial SLAM systems [28,22,31], yet this topic is much less explored in learning-based depth and ego-motion estimation. Though recently IMU has been introduced into both supervised [5,4] and unsupervised [16,34,40] odometry learning, most methods extract IMU features implicitly, while we explicitly utilize IMU dynamics to derive explicit supervisory signals. Li et al. [23] and Wagstaff et al. [37] similarly use EKF for odometry learning. Ours differs in that we do not require ground-truth information [23] or an initialization step [37] to align the velocities and gravities, but learn these quantities using networks. Instead of expressing the error states in the IMU frame, we further derive a camera-centric EKF framework to facilitate the training process. In addition, compared with odometry methods that do not consider the requirements for depth estimation, we specifically design the losses to provide dense depth supervision for monocular depth estimation.

## 3 Methodology

We present the technical details of DynaDepth in this section. We first revisit the preliminaries of IMU motion dynamics. Then we give the details of camera-centric IMU preintegration and the two IMU-related losses, i.e., the scale-aware IMU photometric loss and the cross-sensor photometric consistency loss. Finally, we present the differentiable camera-centric EKF framework which fuses IMU and

camera predictions based on their uncertainties and complements the limitations of each other. A discussion on the connection between DynaDepth and classical visual-inertial SLAM algorithms is also given to provide further insights.

### 3.1 IMU Motion Dynamics

Let  $\{\mathbf{w}_m^b, \mathbf{a}_m^b\}$  and  $\{\mathbf{w}^b, \mathbf{a}^w\}$  denote the IMU measurements and the underlying vehicle angular and acceleration. The superscript  $b$  and  $w$  denote the vector is expressed in the body (IMU) frame or the world frame, respectively. Then we have  $\mathbf{w}_m^b = \mathbf{w}^b + \mathbf{b}^g + \mathbf{n}^g$  and  $\mathbf{a}_m^b = \mathbf{R}_{bw}(\mathbf{a}^w + \mathbf{g}^w) + \mathbf{b}^a + \mathbf{n}^a$ , where  $\mathbf{g}^w$  is the gravity in the world frame and  $\mathbf{R}_{bw}$  is the rotation matrix from the world frame to the body frame [17].  $\{\mathbf{b}^g, \mathbf{b}^a\}$  and  $\{\mathbf{n}^g, \mathbf{n}^a\}$  denote the Gaussian bias and random walk of the gyroscope and the accelerometer, respectively. Let  $\{\mathbf{p}_{wb_t}, \mathbf{q}_{wb_t}\}$  and  $\mathbf{v}_t^w$  denote the translation and rotation from the body frame to the world frame, and the velocity expressed in the world frame at time  $t$ , where  $\mathbf{q}_{wb_t}$  denotes the quaternion. The first-order derivatives of  $\{\mathbf{p}, \mathbf{v}, \mathbf{q}\}$  read:  $\dot{\mathbf{p}}_{wb_t} = \mathbf{v}_t^w$ ,  $\dot{\mathbf{v}}_t^w = \mathbf{a}_t^w$ , and  $\dot{\mathbf{q}}_{wb_t} = \mathbf{q}_{wb_t} \otimes [0, \frac{1}{2}\mathbf{w}^{b_t}]^T$ , where  $\otimes$  denotes the quaternion multiplication. Then the continuous IMU motion dynamics from time  $i$  to  $j$  can be derived as:

$$\mathbf{p}_{wb_j} = \mathbf{p}_{wb_i} + \mathbf{v}_i^w \Delta t + \int \int_{t \in [i, j]} (\mathbf{R}_{wb_t} \mathbf{a}^{b_t} - \mathbf{g}^w) dt^2, \quad (1)$$

$$\mathbf{v}_j^w = \mathbf{v}_i^w + \int_{t \in [i, j]} (\mathbf{R}_{wb_t} \mathbf{a}^{b_t} - \mathbf{g}^w) dt, \quad (2)$$

$$\mathbf{q}_{wb_j} = \int_{t \in [i, j]} \mathbf{q}_{wb_t} \otimes [0, \frac{1}{2}\mathbf{w}^{b_t}]^T dt, \quad (3)$$

where  $\Delta t$  is the time gap between  $i$  and  $j$ . For the discrete cases, we use the averages of  $\{\mathbf{w}, \mathbf{a}\}$  within the time interval to approximate the integrals.

### 3.2 The DynaDepth Framework

DynaDepth aims at jointly training a scale-aware depth network  $\mathcal{M}_d$  and an ego-motion network  $\mathcal{M}_p$  by fusing IMU and camera information. The overall framework is shown in Fig. 1. Given IMU measurements between two consecutive images, we first recover the camera-centric ego-motion  $\{\check{\mathbf{R}}_{c_k c_{k+1}}, \check{\mathbf{p}}_{c_k c_{k+1}}\}$  with absolute scale using IMU motion dynamics, and train two network modules  $\{\mathcal{M}_g, \mathcal{M}_v\}$  to predict the camera-centric gravity and velocity. Then a scale-aware IMU photometric loss and a cross-sensor photometric consistency loss are built based on the ego-motion from IMU. To complement IMU and camera with each other, DynaDepth further integrates a camera-centric EKF module, leading to an updated ego-motion  $\{\hat{\mathbf{R}}_{c_k c_{k+1}}, \hat{\mathbf{p}}_{c_k c_{k+1}}\}$  for the two IMU-related losses.

**IMU Preintegration** IMU usually collects data at a much higher frequency than camera, i.e., between two image frames there exist multiple IMU records.

Since the training losses are defined on ego-motions at the camera frequency, naive use of the IMU motion dynamics requires recalculating the integrals at each training step, which could be computationally expensive. IMU preintegration presents a commonly-used technique to avoid the online integral computation [26,10], which preintegrates the relative pose increment from the IMU records by leveraging the multiplicative property of rotation, i.e.,  $\mathbf{q}_{wb_t} = \mathbf{q}_{wb_i} \otimes \mathbf{q}_{b_ib_t}$ . Then the integration operations can be put into three preintegration terms which only rely on the IMU measurements and can be precomputed beforehand: (1)  $\alpha_{b_ib_j} = \int \int_{t \in [i,j]} (\mathbf{R}_{b_ib_t} \mathbf{a}^{b_t}) dt^2$ , (2)  $\beta_{b_ib_j} = \int_{t \in [i,j]} (\mathbf{R}_{b_ib_t} \mathbf{a}^{b_t}) dt$ , and (3)  $\mathbf{q}_{b_ib_j} = \int_{t \in [i,j]} \mathbf{q}_{b_ib_t} \otimes [0, \frac{1}{2} \mathbf{w}^{b_t}]^T dt$ . Since IMU preintegration is performed in the IMU body frame while the network predicts ego-motions in the camera frame, we thus establish the discrete camera-centric IMU preintegrated ego-motion as:

$$\mathbf{R}_{c_k c_{k+1}}^\sim = \mathbf{R}_{cb} \mathcal{F}^{-1}(\mathbf{q}_{b_k b_{k+1}}) \mathbf{R}_{bc}, \quad (4)$$

$$\mathbf{p}_{c_k c_{k+1}}^\sim = \mathbf{R}_{cb} \alpha_{b_k b_{k+1}} + \mathbf{R}_{c_k c_{k+1}}^\sim \mathbf{R}_{cb} \mathbf{p}_{bc} - \mathbf{R}_{cb} \mathbf{p}_{bc} + \mathbf{v}^{\tilde{c}_k} \Delta t_k - \frac{1}{2} \mathbf{g}^{\tilde{c}_k} \Delta t_k^2, \quad (5)$$

where  $\mathcal{F}$  denotes the transformation from rotation matrix to quaternion.  $\{\mathbf{R}_{cb}, \mathbf{p}_{cb}\}$  and  $\{\mathbf{R}_{bc}, \mathbf{p}_{bc}\}$  are the extrinsics between the IMU and the camera frames. Of note is the estimation of  $\mathbf{v}^{\tilde{c}_k}$  and  $\mathbf{g}^{\tilde{c}_k}$ , which are the velocity and the gravity vectors expressed in the camera frame at time k.

Classical visual-inertial SLAM systems jointly optimize the velocity and the gravity vectors, and accumulate their estimates from previous steps. A complicated initialization step is usually required to achieve good performance. For unsupervised learning where the training units are randomly sampled short-range clips, it is difficult to apply the aforementioned initialization and accumulation. To address this issue, we propose to predict these two quantities directly from images as well during training, using two extra network modules  $\{\mathcal{M}_v, \mathcal{M}_g\}$ .

**IMU Photometric Loss** State-of-the-art visual-inertial SLAM systems usually utilize IMU preintegrated ego-motions by constructing the residues between the IMU preintegrated terms and the system estimates to be optimized. However, naively formulating the training loss as these residues on IMU preintegration terms can only provide sparse supervision for the ego-motion network and thus is inefficient in terms of the entire unsupervised learning system. In this work, we propose an IMU photometric loss  $L_{photo}^{IMU}$  to tackle this problem which provides dense supervisory signals for both the depth and the ego-motion networks. Given an image  $\mathbf{I}$  and its consecutive neighbours  $\{\mathbf{I}_{-1}, \mathbf{I}_1\}$ ,  $L_{photo}^{IMU}$  reads:

$$L_{photo}^{IMU} = \frac{1}{N} \sum_{i=1}^N \min_{\delta \in \{-1, 1\}} \mathcal{L}(\mathbf{I}(\mathbf{y}_i), \mathbf{I}_\delta(\psi(\mathbf{K} \hat{\mathbf{R}}_\delta \mathbf{K}^{-1} \mathbf{y}_i + \frac{\mathbf{K} \hat{\mathbf{p}}_\delta}{\tilde{z}_i}))), \quad (6)$$

$$\mathcal{L}(\mathbf{I}, \mathbf{I}_\delta) = \alpha \frac{1 - SSIM(\mathbf{I}, \mathbf{I}_\delta)}{2} + (1 - \alpha) \|\mathbf{I} - \mathbf{I}_\delta\|_1, \quad (7)$$

where  $\mathbf{K}$  and  $N$  are the camera intrinsics and the number of utilized pixels,  $\mathbf{y}_i$  and  $\tilde{z}_i$  are the pixel coordinate in image  $\mathbf{I}$  and its depth predicted by  $\mathcal{M}_d$ ,  $\mathbf{I}(\mathbf{y}_i)$

is the pixel intensity at  $\mathbf{y}_i$ , and  $\psi(\cdot)$  denotes the depth normalization function.  $\{\hat{\mathbf{R}}_\delta, \hat{\mathbf{p}}_\delta\}$  denotes the ego-motion estimate from image  $\mathbf{I}$  to  $\mathbf{I}_\delta$ , which is obtained by fusing the IMU preintegrated ego-motion and the ones predicted by  $\mathcal{M}_p$  under our camera-centric EKF framework.  $SSIM(\cdot)$  denotes the structural similarity index [39]. We also adopt the per-pixel minimum trick proposed in [14].

**Cross-Sensor Photometric Consistency Loss** In addition to  $L_{photo}^{IMU}$ , we further propose a cross-sensor photometric consistency loss  $L_{photo}^{cons}$  to align the ego-motions from IMU preintegration and  $\mathcal{M}_p$ . Instead of directly comparing the ego-motions, we use the photometric error between the backwarped images, which provides denser supervisory signals for both  $\mathcal{M}_d$  and  $\mathcal{M}_p$ :

$$L_{photo}^{cons} = \frac{1}{N} \sum_{i=1}^N \min_{\delta \in \{-1, 1\}} \mathcal{L}(\mathbf{I}_\delta(\psi(\mathbf{K}\tilde{\mathbf{R}}_\delta\mathbf{K}^{-1}\mathbf{y}_i + \frac{\mathbf{K}\tilde{\mathbf{p}}_\delta}{\tilde{z}_i})), \mathbf{I}_\delta(\psi(\mathbf{K}\hat{\mathbf{R}}_\delta\mathbf{K}^{-1}\mathbf{y}_i + \frac{\mathbf{K}\hat{\mathbf{p}}_\delta}{\hat{z}_i}))), \quad (8)$$

where  $\{\tilde{\mathbf{R}}_\delta, \tilde{\mathbf{p}}_\delta\}$  are the ego-motion predicted by  $\mathcal{M}_p$ .

*Remark:* Of note is that using  $L_{photo}^{cons}$  actually increases the tolerance for illumination change and moving objects which may violate the underlying assumption of the photometric loss between consecutive frames. Since we are comparing two backwarped views in  $L_{photo}^{cons}$ , the errors incurred by the corner cases will be exhibited equally in both backwarped views. In this sense,  $L_{photo}^{cons}$  remains valid, and minimizing  $L_{photo}^{cons}$  helps to align  $\{\tilde{\mathbf{R}}_\delta, \tilde{\mathbf{p}}_\delta\}$  and  $\{\hat{\mathbf{R}}_\delta, \hat{\mathbf{p}}_\delta\}$  under such cases.

**The Camera-Centric EKF Fusion** To fully exploit the complementary IMU and camera sensors, we propose to fuse ego-motions from both sensors under a camera-centric EKF framework. Different from previous methods that integrate EKF into deep learning-based frameworks to deal with IMU data [25,23], ours differs in that we do not require ground-truth ego-motion and velocities to obtain the aligned velocities and gravities for each IMU frame, but propose  $\{\mathcal{M}_v, \mathcal{M}_g\}$  to predict these quantities. In addition, instead of expressing the error states in the IMU body frame, we derive the camera-centric EKF propagation and update processes to facilitate the training process which takes camera images as input.

*EKF Propagation:* Let  $c_k$  denote the camera frame at time  $t_k$ , and  $\{b_t\}$  denote the IMU frames between  $t_k$  and time  $t_{k+1}$  when we receive the next visual measurement. We then propagate the IMU information according to the state transition model:  $\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{w}_t$ , where  $\mathbf{u}_t$  is the IMU record at time  $t$ ,  $\mathbf{w}_t$  is the noise term, and  $\mathbf{x}_t = [\phi_{c_k b_t}^T, \mathbf{p}_{c_k b_t}^T, \mathbf{v}_{c_k}^{c_k T}, \mathbf{g}^{c_k T}, \mathbf{b}_w^{b_t T}, \mathbf{b}_a^{b_t T}]^T$  is the state vector expressed in the camera frame  $c_k$  except for  $\{\mathbf{b}_w, \mathbf{b}_a\}$ .  $\phi_{c_k b_t}$  denotes the so(3) Lie algebra of the rotation matrix  $\mathbf{R}_{c_k b_t}$  s.t.  $\mathbf{R}_{c_k b_t} = \exp([\phi_{c_k b_t}]^\wedge)$ , where  $[\cdot]^\wedge$  denotes the operation from a so(3) vector to the corresponding skew symmetric matrix. To facilitate the derivation of the propagation process, we further separate the state into the nominal states denoted by  $(\cdot)$ , and the error

states  $\delta \mathbf{x}_{b_t} = [\delta \phi_{c_k b_t}^T, \delta \mathbf{p}_{c_k b_t}^T, \delta \mathbf{v}^{c_k T}, \delta \mathbf{g}^{c_k T}, \delta \mathbf{b}_w^{b_t T}, \delta \mathbf{b}_a^{b_t T}]^T$ , such that:

$$\mathbf{R}_{c_k b_t} = \bar{\mathbf{R}}_{c_k b_t} \exp([\delta \phi_{c_k b_t}]^\wedge), \quad \mathbf{p}_{c_k b_t} = \bar{\mathbf{p}}_{c_k b_t} + \delta \mathbf{p}_{c_k b_t}, \quad (9)$$

$$\mathbf{v}^{c_k} = \bar{\mathbf{v}}^{c_k} + \delta \mathbf{v}^{c_k}, \quad \mathbf{g}^{c_k} = \bar{\mathbf{g}}^{c_k} + \delta \mathbf{g}^{c_k}, \quad (10)$$

$$\mathbf{b}_w^{b_t} = \bar{\mathbf{b}}_w^{b_t} + \delta \mathbf{b}_w^{b_t}, \quad \mathbf{b}_a^{b_t} = \bar{\mathbf{b}}_a^{b_t} + \delta \mathbf{b}_a^{b_t}. \quad (11)$$

The nominal states can be computed using the preintegration terms, while the error states are used for propagating the covariances. It is noteworthy that the state transition model of  $\delta \mathbf{x}_{b_t}$  is non-linear, which prevents a naive use of the Kalman filter. EKF addresses this problem and performs propagation by linearizing the state transition model at each time step using the first-order Taylor approximation. Therefore, let  $(\dot{\cdot})$  denote the derivative w.r.t. time  $t$ , we derive the continuous-time propagation model for the error states as:  $\delta \dot{\mathbf{x}}_{b_t} = \mathbf{F} \delta \mathbf{x}_{b_t} + \mathbf{G} \mathbf{n}$ . Detailed derivations are given in the Supplementary material, and  $\mathbf{F}$  and  $\mathbf{G}$  read:

$$\mathbf{F} = \begin{bmatrix} -[\bar{\mathbf{w}}^{b_t}]^\wedge & 0 & 0 & 0 & -\mathbf{I}_3 & 0 \\ 0 & 0 & \mathbf{I}_3 & 0 & 0 & 0 \\ -\bar{\mathbf{R}}_{c_k b_t} [\bar{\mathbf{R}}_{c_k b_t}^T \bar{\mathbf{g}}^{c_k} + \bar{\mathbf{a}}^{b_t}]^\wedge & 0 & 0 & -\mathbf{I}_3 & 0 & -\bar{\mathbf{R}}_{c_k b_t} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} -\mathbf{I}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\bar{\mathbf{R}}_{c_k b_t} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \mathbf{I}_3 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I}_3 \end{bmatrix} \quad (12)$$

where  $\bar{\mathbf{w}}^{b_t} = \mathbf{w}_m^{b_t} - \bar{\mathbf{b}}_w^{b_t}$  and  $\bar{\mathbf{a}}^{b_t} = \mathbf{a}_m^{b_t} - \bar{\mathbf{R}}_{c_k b_t}^T \bar{\mathbf{g}}_{c_k} - \bar{\mathbf{b}}_a^{b_t}$ . Given the continuous error propagation model and the initial condition  $\Phi_{t_\tau, t_\tau} = \mathbf{I}_{18}$ , the discrete state-transition matrix  $\Phi_{(t_{\tau+1}, t_\tau)}$  can be found by solving  $\Phi_{(t_{\tau+1}, t_\tau)} = \mathbf{F}_{t_{\tau+1}} \Phi_{(t_\tau, t_\tau)}$ :

$$\Phi_{t_{\tau+1}, t_\tau} = \exp\left(\int_{t_\tau}^{t_{\tau+1}} \mathbf{F}(s) ds\right) \approx \mathbf{I}_{18} + \mathbf{F} \delta t + \frac{1}{2} \mathbf{F}^2 \delta t^2, \quad \delta t = t_{\tau+1} - t_\tau. \quad (13)$$

Let  $\check{\mathbf{P}}$  and  $\hat{\mathbf{P}}$  denote the prior and posterior covariance estimates during propagation and after an update given new observations. Then we have

$$\mathbf{P}_{t_{\tau+1}}^\check{} = \Phi_{t_{\tau+1}, t_\tau} \mathbf{P}_{t_\tau}^\check{} \Phi_{t_{\tau+1}, t_\tau}^T + \mathbf{Q}_{t_\tau}, \quad (14)$$

$$\mathbf{Q}_{t_\tau} = \int_{t_\tau}^{t_{\tau+1}} \Phi_{s, t_\tau} \mathbf{G} \mathbf{Q} \mathbf{G}^T \Phi_{s, t_\tau}^T ds \approx \Phi_{t_{\tau+1}, t_\tau} \mathbf{G} \mathbf{Q} \mathbf{G}^T \Phi_{t_{\tau+1}, t_\tau}^T \delta t, \quad (15)$$

where  $\mathbf{Q} = \mathcal{D}([\sigma_w^2 \mathbf{I}_3, \sigma_{b_w}^2 \mathbf{I}_3, \sigma_a^2 \mathbf{I}_3, \sigma_{b_a}^2 \mathbf{I}_3])$ .  $\mathcal{D}$  is the diagonalization function.

*EKF Update:* In general, given an observation measurement  $\xi_{k+1}$  and its corresponding covariance  $\mathbf{I}_{k+1}$  from the camera sensor at time  $t_{k+1}$ , we assume the following observation model:  $\xi_{k+1} = h(\mathbf{x}_{k+1}) + \mathbf{n}_r$ ,  $\mathbf{n}_r \sim N(0, \mathbf{I}_{k+1})$ .

Let  $\mathbf{H}_{k+1} = \frac{\partial h(\mathbf{x}_{k+1})}{\partial \delta \mathbf{x}_{k+1}}$ . Then the EKF update applies as following:

$$\mathbf{K}_{k+1} = \check{\mathbf{P}}_{k+1} \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \check{\mathbf{P}}_{k+1} \mathbf{H}_{k+1}^T + \mathbf{I}_{k+1})^{-1}, \quad (16)$$

$$\hat{\mathbf{P}}_{k+1} = (\mathbf{I}_{18} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \check{\mathbf{P}}_{k+1}, \quad (17)$$

$$\delta \hat{\mathbf{x}}_{k+1} = \mathbf{K}_{k+1} (\xi_{k+1} - h(\check{\mathbf{x}}_{k+1})). \quad (18)$$



In DynaDepth, the observation measurement is defined as the ego-motion predicted by  $\mathcal{M}_p$ , i.e.,  $\xi_{k+1} = [\tilde{\phi}_{c_k c_{k+1}}^T, \tilde{p}_{c_k c_{k+1}}^T]^T$ . Of note is that the covariances  $\Gamma_{k+1}$  of  $\{\tilde{\phi}_{c_k c_{k+1}}^T, \tilde{p}_{c_k c_{k+1}}^T\}$  are also predicted by the ego-motion network  $\mathcal{M}_p$ . To finish the camera-centric EKF update step, we derive  $h(\tilde{x}_{k+1})$  and  $\mathbf{H}_{k+1}$  as:

$$h(\tilde{x}_{k+1}) = \begin{bmatrix} \bar{\phi}_{c_k c_{k+1}} \\ \bar{\mathbf{R}}_{c_k b_{k+1}} \mathbf{p}_{bc} + \bar{\mathbf{p}}_{c_k b_{k+1}} \end{bmatrix}, \mathbf{H}_{k+1} = \begin{bmatrix} J_l(-\bar{\phi}_{c_k c_{k+1}})^{-1} \mathbf{R}_{cb} & 0 & 0 & 0 & 0 \\ -\bar{\mathbf{R}}_{c_k b_{k+1}} [\mathbf{p}_{bc}]^\wedge & \mathbf{I}_3 & 0 & 0 & 0 \end{bmatrix}. \quad (19)$$

After obtaining the updated error states  $\delta \hat{\mathbf{x}}_{k+1}$ , we add  $\delta \hat{\mathbf{x}}_{k+1}$  back to the accumulated nominal states to get the corrected ego-motion. In detail,  $\delta \hat{\mathbf{x}}_{k+1}$  is obtained by inserting Eq. (19) into Eq. (16-18), which can be inserted into Eq. (9) to get the updated  $\{\hat{\phi}_{c_k b_{k+1}}, \hat{p}_{c_k b_{k+1}}\}$ . Then by projecting  $\{\hat{\phi}_{c_k b_{k+1}}, \hat{p}_{c_k b_{k+1}}\}$  using the camera intrinsics, we obtain the corrected ego-motion  $\{\hat{\phi}_{c_k b_{k+1}}, \hat{p}_{c_k b_{k+1}}\}$  that fuses IMU and camera information based on their covariances as confidence indicators, which are used to compute  $L_{photo}^{IMU}$  and  $L_{photo}^{cons}$ .

Finally, in addition to  $\{L_{photo}^{IMU}, L_{photo}^{cons}\}$ , the total training loss  $L_{total}$  in DynaDepth also includes the vision-based photometric loss  $L_{photo}^{vis}$  and the disparity smoothness loss  $L_s$  as proposed in monodepth2 [14] to leverage the visual clues. We also consider the weak L2-norm loss  $L_{vg}$  for the velocity and gravity predictions from  $\mathcal{M}_v$  and  $\mathcal{M}_g$ . In summary,  $L_{total}$  reads:

$$L_{total} = L_{photo}^{vis} + \lambda_1 L_s + \lambda_2 L_{photo}^{IMU} + \lambda_3 L_{photo}^{cons} + \lambda_4 L_{vg}, \quad (20)$$

where  $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$  denote the loss weights which are determined empirically.

*Remark:* Although we have witnessed a paradigm shift from EKF to optimization in classical visual-inertial SLAM systems in recent years [28,22,31], we argue that in the setting of unsupervised depth estimation, EKF provides a better choice than optimization. The major problem of EKF is its limited ability to handle long-term data because of the Markov assumption between updates, the first-order approximation for the non-linear state-transition and observation models, and the memory consumption for storing the covariances. However, in our setting, short-term image clips are usually used as the basic training unit, which indicates that the Markov property and the linearization in EKF will approximately hold within the short time intervals. In addition, only the ego-motions predicted by  $\mathcal{M}_p$  are used as the visual measurements, which is memory-efficient.

On the other hand, by using EKF, we are able to correct the IMU preintegrated ego-motions and update  $\{L_{photo}^{IMU}, L_{photo}^{cons}\}$  accordingly when observing new visual measurements. Compared with formulating the commonly-used optimization objective, i.e., the residues of the IMU preintegration terms, as the training losses, our proposed  $L_{photo}^{IMU}$  and  $L_{photo}^{cons}$  provide denser supervision for both  $\mathcal{M}_d$  and  $\mathcal{M}_p$ . From another perspective, EKF essentially can be regarded as weighting the ego-motions from IMU and vision based on their covariances, and thus naturally provides a framework for estimating the uncertainty of the ego-motion predicted by  $\mathcal{M}_p$ , which is non-trivial for the unsupervised learning frameworks.

Table 1: Per-image rescaled depth evaluation on KITTI using the Eigen split. The best and the second best results are shown in **bold** and underline. <sup>†</sup> denotes our reproduced results. Results are rescaled using the median ground-truth from Lidar. The means and standard errors of the scaling ratios are reported in Scale.

Methods	Year	Scale	Error↓				Accuracy↑		
			AbsRel	SqRel	RMSE	RMSE <sub>log</sub>	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Monodepth2 R18 [14]	ICCV 2019	NA	0.112	0.851	4.754	0.190	0.881	0.960	<u>0.981</u>
Monodepth2 R50 <sup>†</sup> [14]	ICCV 2019	29.128±0.084	0.111	0.806	4.642	0.189	0.882	<b>0.962</b>	<b>0.982</b>
PackNet-SfM [15]	CVPR 2020	NA	0.111	0.785	<b>4.601</b>	0.189	0.878	0.960	<b>0.982</b>
Johnston R18 [18]	CVPR 2020	NA	0.111	0.941	4.817	0.189	<b>0.885</b>	<u>0.961</u>	<u>0.981</u>
R-MSFM6 [49]	ICCV 2021	NA	0.112	0.806	4.704	0.191	0.878	0.960	<u>0.981</u>
G2S R50 [3]	ICRA 2021	1.031±0.073	0.112	0.894	4.852	0.192	0.877	0.958	<u>0.981</u>
ScaleInvariant R18 [38]	ICCV 2021	NA	<u>0.109</u>	<u>0.779</u>	4.641	<b>0.186</b>	<u>0.883</u>	<b>0.962</b>	<b>0.982</b>
DynaDepth R18	2022	<u>1.021±0.069</u>	0.111	0.806	4.777	0.190	0.878	0.960	<b>0.982</b>
DynaDepth R50	2022	<b>1.013±0.071</b>	<b>0.108</b>	<b>0.761</b>	<u>4.608</u>	<u>0.187</u>	<u>0.883</u>	<b>0.962</b>	<b>0.982</b>

## 4 Experiment

We evaluate the effectiveness of DynaDepth on KITTI [12] and test the generalization ability on Make3D [33]. In addition, we perform extensive ablation studies on our proposed IMU losses, the EKF framework, the learnt ego-motion uncertainty, and the robustness against illumination change and moving objects.

### 4.1 Implementation

DynaDepth is implemented in pytorch [30]. We adopt the monodepth2 [14] network structures for  $\{\mathcal{M}_d, \mathcal{M}_p\}$ , except that we increase the output dimension of  $\mathcal{M}_p$  from 6 to 12 to include the uncertainty predictions.  $\{\mathcal{M}_g, \mathcal{M}_v\}$  share the same network structure as  $\mathcal{M}_p$  except that the output dimensions are both set to 3.  $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$  are set to  $\{0.001, 0.5, 0.01, 0.001\}$ . We train all networks for 30 epochs using an initial learning rate  $1e-4$ , which is reduced to  $1e-5$  after the first 15 epochs. The training process takes 1 ~ 2 days on a single NVIDIA V100 GPU. The source codes and the trained models will be released.

### 4.2 Scale-Aware Depth Estimation on KITTI

We use the Eigen split [6] for depth evaluation. In addition to the removal of static frames as proposed in [48], we discard images without the corresponding IMU records, leading to 38,102 image-and-IMU triplets for training and 4,238 for validation. WLOG, we use the image resolution 640x192 and cap the depth predictions at 80m, following the common practice in [14,18,15,3,38].

We compare DynaDepth with state-of-the-art monocular depth estimation methods in Table 1, which rescale the results using the ratio of the median depth between the ground-truth and the prediction. For a fair comparison, we only present results achieved with image resolution 640x192 and an encoder with moderate size, i.e., ResNet18 (R18) or ResNet50 (R50). In addition to standard

Table 2: Unscaled depth evaluation on KITTI using the Eigen split. <sup>†</sup> denotes our reproduced results. The best results are shown in **bold**.

Methods	Year	Error↓				Accuracy↑		
		AbsRel	SqRel	RMSE	RMSE <sub>log</sub>	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Monodepth2 R50 <sup>†</sup> [14]	ICCV 2019	0.966	15.039	19.145	3.404	0.000	0.000	0.000
PackNet-SfM [15]	CVPR 2020	0.111	0.829	4.788	0.199	0.864	0.954	0.980
G2S R50 [3]	ICRA 2021	<b>0.109</b>	0.860	4.855	0.198	0.865	0.954	0.980
DynaDepth R50	2022	<b>0.109</b>	<b>0.787</b>	<b>4.705</b>	<b>0.195</b>	<b>0.869</b>	<b>0.958</b>	<b>0.981</b>

depth evaluation metrics [7], we report the means and standard errors of the rescaling factors to demonstrate the scale-awareness ability. DynaDepth achieves the best up-to-scale performance w.r.t. four metrics and achieves the second best for the other three metrics. Of note is that DynaDepth also achieves a nearly perfect absolute scale. In terms of scale-awareness, even our R18 version outperforms G2S R50 [3], which uses a heavier encoder. For better illustration, we also show the scaling ratio histograms with and without IMU in Fig. 1(b).

We then report the unscaled results in Table 2, and compare with PackNet-SfM [15] and G2S [3], which use the GPS information to construct velocity constraints. Without rescaling, Monodepth2 [14] fails completely as expected. In this case, DynaDepth achieves the best performance w.r.t. all metrics, setting a new benchmark of unscaled depth evaluation for monocular methods.

### 4.3 Generalizability on Make3D

We further test the generalizability of DynaDepth on Make3D [33] using models trained on KITTI [12]. The test images are centre-cropped to a 2x1 ratio for a fair comparison with previous methods [14]. A qualitative example is given in Fig. 1(c), where the model without IMU fails in the glass and shadow areas, while our model achieves a distinguishable prediction. Quantitative results are reported in Table 3. A reasonably good scaling ratio has been achieved for DynaDepth, indicating that the scale-awareness learnt by DynaDepth can be well generalized to unseen datasets. Surprisingly, we found that DynaDepth that only uses the gyroscope and accelerator IMU information (w.o/  $L_{vg}$ ) achieves the best generalization results. The reason can be two-fold. First, our full model may overfit to the KITTI dataset due to the increased modeling capacity. Second, the performance degradation can be due to the domain gap of the visual data, since both  $\mathcal{M}_v$  and  $\mathcal{M}_g$  take images as input. This also explains the scale loss of G2S in this case. We further show that DynaDepth w.o/  $L_{vg}$  significantly outperforms the stereo version of Monodepth2, which can also be explained by the visual domain gap, especially the different camera intrinsics used in their left-right consistency loss. Our generalizability experiment justifies the advantages of using IMU to provide scale information, which will not be affected by the visual domain gap and varied camera parameters, leading to improved generalization performance. In addition, it is also shown that the use of EKF in training significantly improves

Table 3: Generalization results on Make3D. \* denotes unscaled results while the others present per-image rescaled results. The best results are shown in **bold**. M, S, GPS, and IMU in Type denote whether monocular, stereo, GPS and IMU information are used for training the model on KITTI. - means item not available.

Methods	$L_{vg}$	EKF	Type	Scale	Error↓				Accuracy↑		
					Abs <sub>rel</sub>	Sq <sub>rel</sub>	RMSE	RMSE <sub>log</sub>	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Zhou [48]	-	-	M	-	0.383	5.321	10.470	0.478	-	-	-
Monodepth2 [14]	-	-	M	-	0.322	3.589	7.417	0.163	-	-	-
G2S [3]	-	-	M+GPS	2.81±0.85	-	-	-	-	-	-	-
DynaDepth			M+IMU	1.37±0.27	0.316	3.006	7.218	0.164	0.522	0.797	0.914
DynaDepth		✓	M+IMU	<b>1.26±0.27</b>	<b>0.313</b>	<b>2.878</b>	<b>7.133</b>	<b>0.162</b>	<b>0.527</b>	<b>0.800</b>	<b>0.916</b>
DynaDepth (full)	✓	✓	M+IMU	1.45± <b>0.26</b>	0.334	3.311	7.463	0.169	0.497	0.779	0.908
Monodepth2* [14]	-	-	M+S	-	0.374	3.792	8.238	<b>0.201</b>	-	-	-
DynaDepth*			M+IMU	-	0.360	3.461	8.833	0.226	0.295	0.594	0.794
DynaDepth*		✓	M+IMU	-	<b>0.337</b>	<b>3.135</b>	<b>8.217</b>	<b>0.201</b>	<b>0.384</b>	<b>0.671</b>	<b>0.845</b>
DynaDepth* (full)	✓	✓	M+IMU	-	0.378	3.655	9.034	0.240	0.261	0.550	0.758

the generalization ability, possibly thanks to the EKF fusion framework that takes the uncertainty into account and integrates the generalizable IMU motion dynamics and the domain-specific vision information in a more reasonable way.

#### 4.4 Ablation Studies

We conduct ablation studies on KITTI to investigate the effects of the proposed IMU-related losses, the EKF fusion framework, and the learnt ego-motion uncertainty. In addition, we design simulated experiment to demonstrate the robustness of DynaDepth against vision degradation such as illumination change and moving objects. WLOG, we use ResNet18 as the encoder for all ablation studies.

##### The effects of the IMU-related losses and the EKF Fusion Framework

We report the ablation results of the IMU-related losses and the EKF fusion framework in Table 4. First,  $L_{photo}^{IMU}$  presents the main contributor to learning the scale. However, only a rough scale is learnt using  $L_{photo}^{IMU}$  only. And the up-to-scale accuracy is also not as good as the other models.  $L_{photo}^{cons}$  provides better up-to-scale accuracy, but using  $L_{photo}^{cons}$  alone is not enough to learn the absolute scale due to the relatively weak supervision. Instead, combining  $L_{photo}^{IMU}$  and  $L_{photo}^{cons}$  together boosts the performance of both the scale-awareness and the accuracy. The use of  $L_{vg}$  further enhances the evaluation results. Nevertheless, as shown in Section 4.3,  $L_{vg}$  may lead to overfitting to current dataset and harm the generalizability, due to its dependence on visual data that suffers from the visual domain gap between different datasets. On the other hand, EKF improves the up-to-scale accuracy w.r.t. almost all metrics, while decreasing the learnt scale information a little bit. Since the scale information comes from IMU, and the visual data contributes most to the up-to-scale accuracy, EKF achieves a good balance between the two sensors. Moreover, as shown in Table 3, the use of EKF leads to the best generalization results w.r.t. both the scale and the accuracy.

Table 4: Ablation results of the IMU-related losses and the EKF fusion framework on KITTI. The best results are shown in **bold**.

EKF	$L_{photo}^{IMU}$	$L_{photo}^{cons}$	$L_{vg}$	Scale	Error↓				Accuracy↑		
					AbsRel	SqRel	RMSE	RMSE <sub>log</sub>	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
✓	✓			1.130±0.099	0.115	0.804	4.806	0.193	0.871	0.959	<b>0.982</b>
✓		✓		4.271±0.089	0.114	0.832	4.780	0.192	0.876	0.959	0.981
✓	✓	✓		1.076±0.095	0.113	<b>0.794</b>	<b>4.760</b>	0.191	0.874	<b>0.960</b>	<b>0.982</b>
✓	✓	✓	✓	<b>1.021±0.069</b>	<b>0.111</b>	0.806	4.777	<b>0.190</b>	<b>0.878</b>	<b>0.960</b>	<b>0.982</b>
	✓	✓		<b>0.968±0.098</b>	0.115	0.839	4.898	0.194	0.869	0.958	0.981
✓	✓	✓		1.076±0.095	<b>0.113</b>	<b>0.794</b>	<b>4.760</b>	<b>0.191</b>	<b>0.874</b>	<b>0.960</b>	<b>0.982</b>
	✓	✓	✓	<b>1.013±0.069</b>	0.112	0.808	<b>4.751</b>	0.191	0.877	<b>0.960</b>	<b>0.982</b>
✓	✓	✓	✓	<b>1.021±0.069</b>	<b>0.111</b>	<b>0.806</b>	4.777	<b>0.190</b>	<b>0.878</b>	<b>0.960</b>	<b>0.982</b>

**The robustness against vision degradation** We then examine the robustness of DynaDepth against illumination change and moving objects, two major cases that violate the underlying assumption of the photometric loss. We simulate the illumination change by randomly alternating image contrast within a range 0.5. The moving objects are simulated by randomly inserting three 150x150 black squares. In contrast to data augmentation, we perform the perturbation for each image independently, rather than applying the same perturbation to all images in a triplet. Results are given in Table 5. Under illumination change, the accuracy of Monodepth2 degrades as expected, while DynaDepth rescues the accuracy to a certain degree and maintains the correct absolute scales. EKF improves almost all metrics in this case, and using both EKF and  $L_{vg}$  achieves the best scale and AbsRel. However, the model without  $L_{vg}$  obtains the best performance on most metrics. The reason may be the dependence of  $L_{vg}$  on the visual data, which is more sensitive to image qualities. When there exist moving objects, Monodepth2 fails completely. Using DynaDepth without EKF and  $L_{vg}$  improves the up-to-scale accuracy a little bit, but the results are still far from expected. Using EKF significantly improves the up-to-scale results, while it is still hard to learn the scale given the difficulty of the task. In this case, using  $L_{vg}$  is shown to provide strong scale supervision and achieve a good scale result.

**The learnt ego-motion uncertainty** We illustrate the training progress of the ego-motion uncertainty in Fig. 2. We report the averaged covariance as the uncertainty measure. The learnt uncertainty exhibits a similar pattern as the depth error (AbsRel), meaning that the model becomes more certain about its predictions as the training continues. Of note is that only indirect supervision is provided, which justifies the effectiveness of our fusion framework. In addition, DynaDepth R50 achieves a lower uncertainty than R18, indicating that a larger model capacity also contributes to the prediction confidence, yet such difference can hardly be seen w.r.t. AbsRel. Table 6 presents another interesting observation. In KITTI, the axis-z denotes the forward direction. Since most test images correspond to driving forward, the magnitude of  $t_z$  is significantly larger than

Table 5: Ablation results of the robustness against vision degradation on the simulated data from KITTI. The best results are shown in **bold**. IC and MO denote the two investigated vision degradation types, i.e., illumination change and moving objects. - means item not available. <sup>†</sup> denotes our reproduced results.

Methods	EKF	$L_{vg}$	Type	Scale	Error↓				Accuracy↑		
					AbsRel	SqRel	RMSE	RMSE <sub>log</sub>	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Monodepth2 <sup>†</sup> [14]	-	-	IC	27.701±0.096	0.127	0.976	5.019	0.220	0.855	0.946	0.972
DynaDepth			IC	1.036±0.099	0.124	<b>0.858</b>	4.915	0.226	0.852	0.950	0.977
DynaDepth	✓		IC	0.946±0.089	0.123	0.925	<b>4.866</b>	<b>0.196</b>	<b>0.863</b>	<b>0.957</b>	<b>0.981</b>
DynaDepth	✓	✓	IC	<b>1.019±0.074</b>	<b>0.121</b>	0.906	4.950	0.217	0.859	0.954	0.978
Monodepth2 <sup>†</sup> [14]	-	-	MO	0.291±0.176	0.257	2.493	8.670	0.398	0.584	0.801	0.897
DynaDepth			MO	0.083±0.225	0.169	1.290	6.030	0.278	0.763	0.915	0.960
DynaDepth	✓		MO	0.087±0.119	0.126	<b>0.861</b>	5.312	<b>0.210</b>	0.840	0.948	<b>0.979</b>
DynaDepth	✓	✓	MO	<b>0.956±0.084</b>	<b>0.125</b>	0.926	<b>4.954</b>	0.214	<b>0.852</b>	<b>0.949</b>	0.976

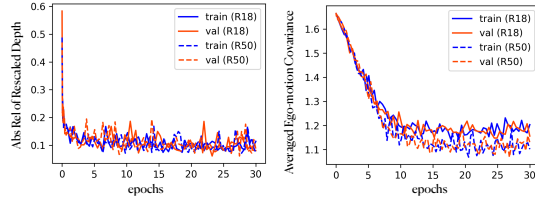


Fig. 2: The training processes w.r.t. AbsRel (left) and the averaged ego-motion covariance (right).

Table 6: The averaged magnitude  $|\bar{t}|$  and the variance  $\bar{\sigma}_t^2$  of the translation predictions along each axis.

	axis-x	axis-y	axis-z
$ \bar{t} $	0.017	0.018	0.811
$\bar{\sigma}_t^2$	7.559	5.222	0.105

$\{t_x, t_y\}$ . Accordingly, DynaDepth shows a high confidence on  $t_z$ , while large variances are observed for  $\{t_x, t_y\}$ , potentially due to the difficulty to distinguish the noises from the small amount of translations along axis-x and axis-y.

## 5 Conclusion

In this paper, we propose DynaDepth, a scale-aware, robust, and generalizable monocular depth estimation framework using IMU motion dynamics. Specifically, we propose an IMU photometric loss and a cross-sensor photometric consistency loss to provide dense supervision and absolute scales. In addition, we derive a camera-centric EKF framework for the sensor fusion, which also provides an ego-motion uncertainty measure under the setting of unsupervised learning. Extensive experiments support that DynaDepth is advantageous w.r.t. learning absolute scales, the generalizability, and the robustness against vision degradation.

**Acknowledgment** This work is supported by ARC FL-170100117, DP-180103424, IC-190100031, and LE-200100049.

## References

1. Bhat, S.F., Alhashim, I., Wonka, P.: Adabins: Depth estimation using adaptive bins. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4009–4018 (2021)
2. Bian, J., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M.M., Reid, I.: Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in Neural Information Processing Systems* **32** (2019)
3. Chawla, H., Varma, A., Arani, E., Zonooz, B.: Multimodal scale consistency and awareness for monocular self-supervised depth estimation. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 5140–5146. IEEE (2021)
4. Chen, C., Rosa, S., Miao, Y., Lu, C.X., Wu, W., Markham, A., Trigoni, N.: Selective sensor fusion for neural visual-inertial odometry. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 10542–10551 (2019)
5. Clark, R., Wang, S., Wen, H., Markham, A., Trigoni, N.: Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 31 (2017)
6. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 2650–2658 (2015)
7. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems* **27** (2014)
8. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(3), 611–625 (2017)
9. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 834–849. Springer (2014)
10. Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D.: Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. *Georgia Institute of Technology* (2015)
11. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2002–2011 (2018)
12. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* **32**(11), 1231–1237 (2013)
13. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 270–279 (2017)
14. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 3828–3838 (2019)
15. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3d packing for self-supervised monocular depth estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 2485–2494 (2020)
16. Han, L., Lin, Y., Du, G., Lian, S.: Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 6906–6913. IEEE (2019)

17. Huang, G.: Visual-inertial navigation: A concise review. In: 2019 IEEE International Conference on Robotics and Automation (ICRA). pp. 9572–9582. IEEE (2019)
18. Johnston, A., Carneiro, G.: Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4756–4765 (2020)
19. Jung, H., Park, E., Yoo, S.: Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12642–12652 (2021)
20. Khan, F., Salahuddin, S., Javidnia, H.: Deep learning-based monocular depth estimation methods—a state-of-the-art review. *Sensors* **20**(8), 2272 (2020)
21. Klodt, M., Vedaldi, A.: Supervising the new with the old: learning sfm from sfm. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 698–713 (2018)
22. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual-inertial odometry using nonlinear optimization. *The International Journal of Robotics Research* **34**(3), 314–334 (2015)
23. Li, C., Waslander, S.L.: Towards end-to-end learning of visual inertial odometry with an ekf. In: 2020 17th Conference on Computer and Robot Vision (CRV). pp. 190–197. IEEE (2020)
24. 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* **38**(10), 2024–2039 (2015)
25. Liu, W., Caruso, D., Ilg, E., Dong, J., Mourikis, A.I., Daniilidis, K., Kumar, V., Engel, J.: Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters* **5**(4), 5653–5660 (2020)
26. Lupton, T., Sukkarieh, S.: Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics* **28**(1), 61–76 (2011)
27. Mahjourian, R., Wicke, M., Angelova, A.: 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. pp. 5667–5675 (2018)
28. Mourikis, A.I., Roumeliotis, S.I., et al.: A multi-state constraint kalman filter for vision-aided inertial navigation. In: 2007 IEEE International Conference on Robotics and Automation (ICRA). vol. 2, p. 6 (2007)
29. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics* **31**(5), 1147–1163 (2015)
30. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems* **32** (2019)
31. Qin, T., Li, P., Shen, S.: Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics* **34**(4), 1004–1020 (2018)
32. Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., Black, M.J.: Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12240–12249 (2019)
33. 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* **31**(5), 824–840 (2008)



34. Shamwell, E.J., Lindgren, K., Leung, S., Nothwang, W.D.: Unsupervised deep visual-inertial odometry with online error correction for rgb-d imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **42**(10), 2478–2493 (2019)
35. Shu, C., Yu, K., Duan, Z., Yang, K.: Feature-metric loss for self-supervised learning of depth and egomotion. In: *European Conference on Computer Vision*. pp. 572–588. Springer (2020)
36. Taketomi, T., Uchiyama, H., Ikeda, S.: Visual slam algorithms: A survey from 2010 to 2016. *IPSN Transactions on Computer Vision and Applications* **9**(1), 1–11 (2017)
37. Wagstaff, B., Wise, E., Kelly, J.: A self-supervised, differentiable Kalman filter for uncertainty-aware visual-inertial odometry. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (2022)
38. Wang, L., Wang, Y., Wang, L., Zhan, Y., Wang, Y., Lu, H.: Can scale-consistent monocular depth be learned in a self-supervised scale-invariant manner? In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 12727–12736 (2021)
39. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13**(4), 600–612 (2004)
40. Wei, P., Hua, G., Huang, W., Meng, F., Liu, H.: Unsupervised monocular visual-inertial odometry network. In: *Proceedings of the Twenty-Ninth International Conference on Artificial Intelligence*. pp. 2347–2354 (2021)
41. Yang, N., Stumberg, L.v., Wang, R., Cremers, D.: 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*. pp. 1281–1292 (2020)
42. Yin, Z., Shi, J.: Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1983–1992 (2018)
43. Zhan, H., Weerasekera, C.S., Bian, J.W., Reid, I.: Visual odometry revisited: What should be learnt? In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 4203–4210. IEEE (2020)
44. Zhang, J., Tao, D.: Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal* **8**(10), 7789–7817 (2020)
45. Zhang, S., Zhang, J., Tao, D.: Information-theoretic odometry learning. *arXiv preprint arXiv:2203.05724* (2022)
46. Zhang, S., Zhang, J., Tao, D.: Towards scale consistent monocular visual odometry by learning from the virtual world. *arXiv preprint arXiv:2203.05712* (2022)
47. Zhao, W., Liu, S., Shu, Y., Liu, Y.J.: Towards better generalization: Joint depth-pose learning without posenet. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 9151–9161 (2020)
48. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1851–1858 (2017)
49. Zhou, Z., Fan, X., Shi, P., Xin, Y.: R-msfm: Recurrent multi-scale feature modulation for monocular depth estimating. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 12777–12786 (2021)