AdaDepth: Exploiting Inherent Scene Information for Self-Supervised Depth Estimation in Dynamic Scenes

Xuanang Gao[1], Xiongbin Wu[1], Zhiwei Ning[1], Runze Yang[1, 2], Zhonglong Zheng[3*], Jie Yang[1*], Wei Liu[1*]

[1] School of Automation and Intelligent Sensing, Shanghai Jiao Tong University
[2]School of Computing, Macquarie University
[3]School of Computer Science, Zhejiang Normal University
{fangkuar, pokerjoker, zwning, runze.y, jieyang, weiliucv}@sjtu.edu.cn
zhonglong@zjnu.edu.cn

Abstract

Self-supervised monocular depth estimation methods severely compromise accuracy in dynamic objects due to their static scene assumption. Existing approaches for dynamic scenes suffer from two critical shortcomings: 1) reliance on supervised segmentation models (requiring costly annotations) or computationally intensive multi-branch models to isolate moving objects, and 2) simple integration of 2D/3D motion flow without reliable supervision for dynamic objects. We propose AdaDepth, a two-stage framework that jointly performs unsupervised scene decomposition and dynamic-aware depth learning. In the initial structural stage, our geometry-motion joint scene decomposition (GMoDecomp) module ensures the robust generation of a depth prior and simultaneously partitions the scene into multiple regions through the fusion of geometric and motion cues. In the region-adaptive refinement stage, we exploit the depth prior and decomposed regions to introduce motion-aware and geometry-consistent constraints, effectively improving depth estimation in dynamic scenes. AdaDepth achieves accurate depth prediction in highly dynamic scenes without relying on external labels or specialized segmentation models. Extensive experiments on KITTI, Cityscapes, and Waymo Open demonstrate its superiority over state-of-the-art approaches.

Code — https://github.com/xagao/AdaDepth

Introduction

Monocular depth estimation (MDE) is a critical component in computer vision, which finds applications in autonomous driving, 3D reconstruction, and VR/AR. Traditionally, depth information has been obtained using precise 3D sensors (e.g., LiDAR or structured light). However, these methods can be costly and complex, leading to the development of deducing the depth information of a scene from a single camera. Recently, self-supervised methods (Zhou et al. 2017; Godard et al. 2019; Yin and Shi 2018; Zhang et al. 2023) use unlabeled monocular videos to generate supervision through view synthesis, removing the need for expensive labels.

However, for self-supervised approaches to learn depth, dynamic scenes with moving objects pose a challenge.

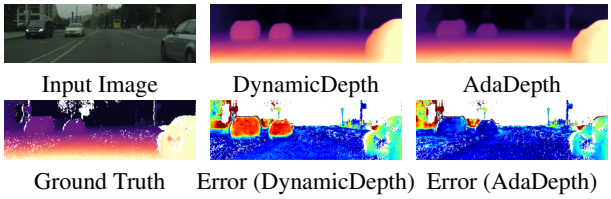

Figure 1: Qualitative comparison of depth predictions between DynamicDepth (Feng et al. 2022) and our AdaDepth. Warmer colors in the error maps indicate larger errors. Our method better captures depth in dynamic regions.

Methods that assume a static scene (Zhou et al. 2017; Godard et al. 2019; Yin and Shi 2018; Bian et al. 2019; Zhou, Greenwood, and Taylor 2021) often produce inaccurate depth estimates in dynamic regions, as the depth is incorrectly adjusted to explain the motion of moving objects. To handle the motion part in a dynamic scene, some methods (Li et al. 2021; Ranjan et al. 2019; Sun and Hariharan 2023) incorporate additional pixel-wise object motion estimation in 2D/3D. Numerous studies on dynamic scene modeling integrate pre-trained segmentation models (Lee et al. 2021a; Feng et al. 2022; Moon et al. 2024) or multi task-specific models (Nguyen et al. 2024) to identify dynamic objects, to enhance the performance.

Despite notable advancements in handling dynamic objects, their limitations highlight the need for further exploration. Entangling object motion and depth estimation introduces inherent ambiguity in jointly learning the depth and motion of the objects without precise supervision. Methods integrating supervised segmentation models fail to distinguish dynamic and static objects and rely on costly labeled data, while those employing multiple task-specific models to identify dynamic objects introduce computational redundancy due to complex pre-training pipelines.

In this paper, to address the challenges above, we propose a novel framework, AdaDepth, for self-supervised depth estimation that significantly improves depth accuracy, particularly in dynamic scenes. Unlike existing methods that rely on pre-computed segmentation masks or multi-task models, AdaDepth decomposes dynamic scenes using our proposed

*corresponding authors.

geometry-motion joint scene decomposition (GMoDecomp) module through inherent stereoscopic perception and motion cues. In the initial structural inference stage, GMoDecomp is jointly optimized with the depth prior network, leveraging the depth information to ensure reliable scene decomposition while simultaneously refining the depth prior through feedback from the decomposition results. For the training in the region-adaptive depth refinement stage, we introduce a motion-robust reprojection loss and a ground-adherent depth constraint to generate a high-quality final depth. These two losses leverage the obtained scene decomposition results and robust depth priors to mitigate the negative effects of imperfect identification and provide reliable supervision in dynamic regions. As illustrated in Fig. 1, our method accurately predicts the depth of dynamic objects. In summary, we present the following contributions:

- We propose a novel framework that significantly improves depth accuracy for dynamic objects. We first introduce GMoDecomp in the first stage, a novel module that achieves dynamic scene decomposition by exploiting the information embedded in the scene, while it simultaneously ensures the rigidity-preserving depth priors.
- We propose a region-adaptive refinement stage that merges the depth prior and scene decomposition with a motion-robust reprojection loss and a ground-adherent depth constraint to mitigate error propagation from misidentification and ensure reliable supervision for dynamic objects.
- Our framework, orthogonal to architecture design, enables seamless integration with existing depth networks to improve their performance. Experiments demonstrate that AdaDepth achieves state-of-the-art results on KITTI, Waymo Open, and Cityscapes datasets.

Related Work

Self-supervised Monocular Depth Estimation

Due to the limited availability of ground-truth depth, self-supervised learning of depth estimation is widely studied (Godard, Mac Aodha, and Brostow 2017; Godard et al. 2019; Casser et al. 2019; Shu et al. 2020; Li et al. 2021). Early methods rely on stereo pairs (Garg et al. 2016), while a later approach (Zhou et al. 2017) relaxes this constraint by jointly estimating depth and ego-motion from monocular videos, optimizing photometric reprojection loss. To further improve the training signal and mitigate occlusion effects, a per-pixel minimum reprojection loss is introduced (Godard et al. 2019). 3D geometric consistency (Mahjourian, Wicke, and Angelova 2018) and depth prediction consistency (Bian et al. 2019) across consecutive frames are proposed. Furthermore, advanced network architectures (Guizilini et al. 2020a; Lyu et al. 2021; Zhou, Greenwood, and Taylor 2021; Han et al. 2022; Han, Yin, and Shen 2023) and the ones that adopt semantic guidance (Jung, Park, and Yoo 2021; Klingner et al. 2020) are proposed. Recently, the methods of taking multiple images as input and utilizing the cost volumes are also introduced (Watson et al. 2021; Guizilini et al. 2022; Woo et al. 2025).

Handling Dynamic Objects in Self-supervised Depth Estimation

Recent approaches leverage a pretrained segmentation network (Feng et al. 2022; Guizilini et al. 2020b; Klingner et al. 2020; Nguyen et al. 2024; Moon et al. 2024; Lee et al. 2021a) to discern all possible dynamic objects (e.g., person, rider, car, truck, and others). To generate better depth in dynamic regions, recent works combine these masks to leverage cycle-consistency loss (Feng et al. 2022), scale-consistent pseudo depth across all possible dynamic objects (Nguyen et al. 2024), and a ground-contacting-prior disparity smoothness loss (Moon et al. 2024) for precise depth supervision. While leveraging a useful off-the-shelf network is effective in identifying moving objects, it comes with several drawbacks, including manually labeled labels for segmentation network training, which is inconsistent with the original intent of self-supervised methods, and the potential inclusion of static objects in segmentation masks. In contrast, our proposed AdaDepth identifies potential moving objects solely based on the provided images and synthesized images, eliminating the requirement for labeled segmentation.

Other works leverage jointly learned optical flow to refine the depth prediction in dynamic scenes. The methods (Vankadari et al. 2023; Yin and Shi 2018) predict a residual optical flow to handle dynamic regions and enforce forward-backward flow consistency (Zou, Luo, and Huang 2018) to reduce the impact on the static areas. In addition, there are also approaches (Luo et al. 2019; Ranjan et al. 2019) that model the dynamic objects by jointly learning depth, ego-motion, and optical flow. Other works propose to model 3D scene flow for dynamic objects to achieve accurate depth. A 3D object motion network is proposed to jointly learn the residual flow field through a sparsity loss (Li et al. 2021), and RM-Depth (Hui 2022) proposes to estimate a 3D motion field of moving objects in a coarse-to-fine framework. Dynamo-Depth (Sun and Hariharan 2023) introduces independent motion networks to generate a complete flow and motion mask for disambiguating dynamical motion. Although these works use sparsity loss (Li et al. 2021), outlier-aware regularization loss (Hui 2022), or a jointly learned motion mask to minimize the impact on the static scenes, ambiguity can still arise during pixel matching in static regions or even in dynamic scenes. Instead of using the 2D/3D flow directly for additional estimation of dynamic objects, we propose to synthesize pseudo-images via an optical flow network, which builds more accurate supervision in dynamic regions and does not affect the depth of static areas.

Method

Self-Supervised Monocular Depth Estimation

This subsection briefly reviews the conventional self-supervised monocular depth estimation pipeline. Given a target image I_t and temporally adjacent source images I_s, a depth network predicts a depth map D_t for I_t. Simultaneously, a pose network estimates the relative 6-DoF camera pose $T_{t\to s}$ from I_t to I_s. The reconstructed view of I_t from the source view I_s can be obtained by:

$$I_{s\to t} = BW(I_s, proj(D_t, T_{t\to s}, K)), \tag{1}$$

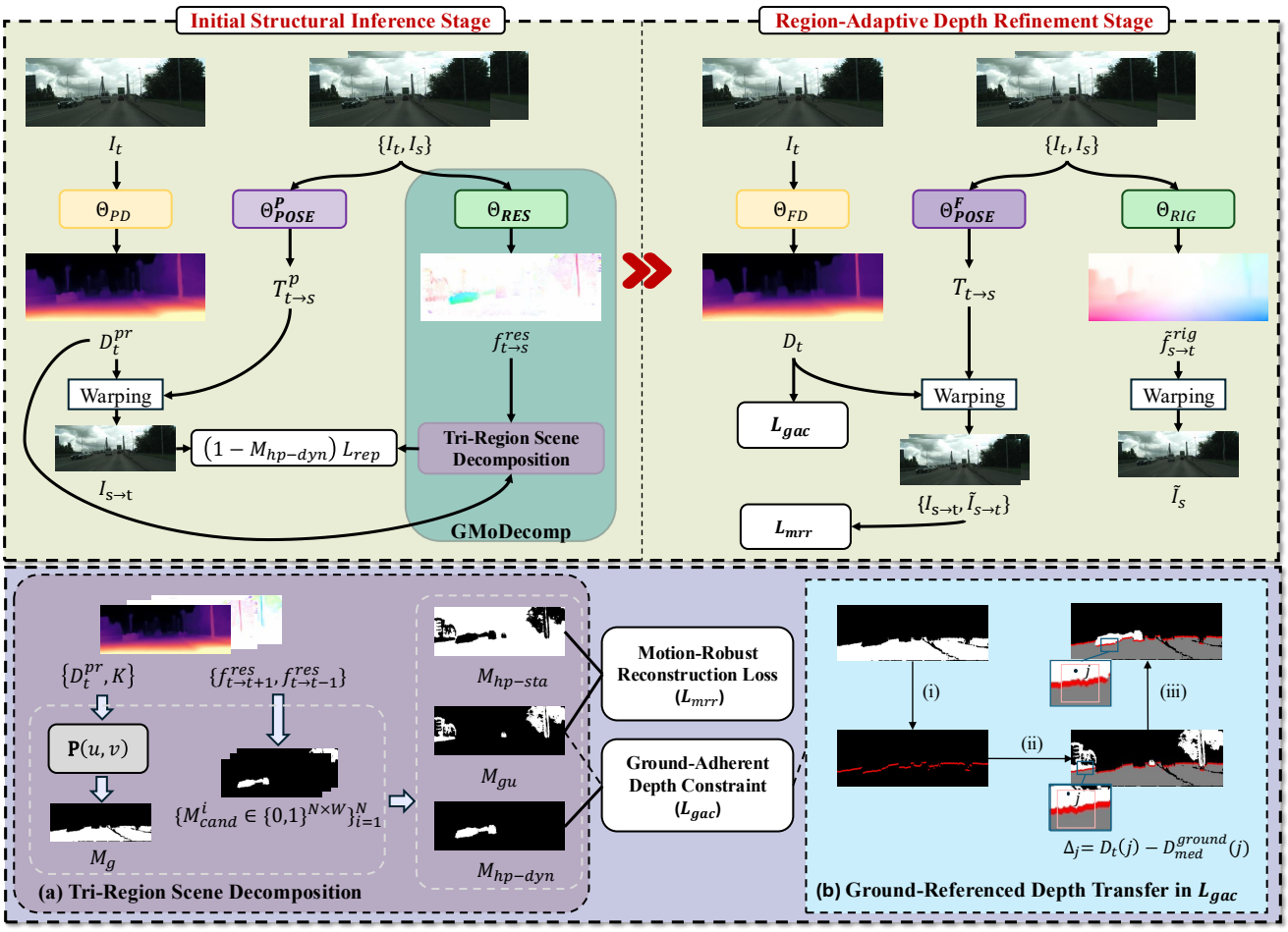

Figure 2: Overview of the proposed framework. In the structural inference stage, our framework integrates a depth prior network Θ_{PD} to predict the depth prior D^{pr}_t and GMoDecomp to decompose the scene using geometry and motion cues, forming a cyclic optimization for robust D^{pr}_t and scene decomposition. In the region-adaptive depth refinement stage, we leverage D^{pr}_t and scene decomposition to form a motion-robust reprojection loss L_{mrr} and a ground-adherent depth constraint L_{gac}, which together mitigate errors from misclassified dynamic pixels and provide reliable supervision for dynamic objects.

where $proj()$ returns the 2D coordinates of the pixels in I_t reprojected to I_s, BW indicates the differentiable backward warping operation (Jaderberg et al. 2015), and K represents the camera intrinsics matrix assumed to be known. Then the reprojection loss L_{rep} defined as the photometric error between $I_{s\to t}$ and I_t, consisting of $L1$ and SSIM (Wang et al. 2004) error terms balanced by α:

$$L_{rep}(I_t, I_{s\to t}) = \alpha \cdot L1(I_t, I_{s\to t}) + (1-\alpha)\cdot SSIM(I_t, I_{s\to t}). \quad (2)$$

In addition, the edge-aware smoothness loss L_{sm} is employed to encourage locally smooth depth maps, defined as:

$$L_{sm}(D_t, I_t) = |\partial_x D^*_t|\, e^{-|\partial_x I_t|} + |\partial_y D^*_t|\, e^{-|\partial_y I_t|}, \quad (3)$$

where $\partial_x D^*_t$ and $\partial_y D^*_t$ are the x/y-axis gradients of the mean-normalized inverse depth $D^*_t = D_t/\bar{D}_t$, which discourages the shrinking of the estimated depth.

Overview

Fig. 2 provides an overview of our framework, which follows a decomposition-to-regularization paradigm based on geometric and motion cues. It comprises two coupled stages: (i) the initial structural inference stage and (ii) the region-adaptive depth refinement stage. In the first stage, we jointly estimate a geometry-consistent depth prior D^{pr}_t and decompose the scene into multiple regions using the proposed GMoDecomp. By leveraging both geometric cues and residual motion estimation, GMoDecomp enables a tri-region decomposition, covering high-probability dynamic, grounded uncertain, and high-probability static regions. In the second stage, we train a final depth network to predict high-quality depth D_t under region-adaptive supervision. It integrates the structural decomposition masks and the depth prior D^{pr}_t to mitigate erroneous gradients from dynamic content and propagate reliable supervision across different regions.

Initial Structural Inference Stage

In this stage, we jointly estimate a depth prior and perform structure-aware scene decomposition. The two predictions are mutually reinforced through joint optimization, leading

to progressively improved depth and decomposition results.

Depth Prior via Masked Photometric Consistency. We first adopt a depth prior network Θ_{PD} and a camera pose network Θ^{P}_{POSE} to predict a depth prior map D^{pr}_t and a camera relative pose $T^{p}_{t\to s}$. To improve depth prior estimation in dynamic scenes, we mask out unreliable regions where photometric consistency fails. Specifically, high-confidence dynamic pixels, identified by the binary mask $M_{hp\text{-}dyn}$ from GMoDecomp (see below) are excluded from supervision. The masked reprojection loss is defined as:

$$L^{m}_{rep} = (1 - M_{hp\text{-}dyn}) \cdot L_{rep}. \tag{4}$$

GMoDecomp: Geometry-Motion Joint Scene Decomposition Module. To disentangle the moving object motion from camera-induced flow and achieve robust scene decomposition, we introduce GMoDecomp, a module that first captures residual motion and fuses it with geometric constraints to derive a tri-region segmentation of the scene.

Residual Flow Estimation as Motion Cues. To capture object-specific non-rigid motion, we employ a lightweight residual motion network Θ_{RES} to estimate the residual flow $f^{res}_{t\to s}$. Combined with the rigid flow computed from D^{pr}_t and pose $T^{p}_{t\to s}$:

$$f^{rig}_{t\to s}(p_t) = proj(D^{pr}_t(p_t), T^{p}_{t\to s}, K) - p_t, \tag{5}$$

the full flow is $f^{full}_{t\to s} = f^{rig}_{t\to s} + f^{res}_{t\to s}$. Θ_{RES} is supervised via reprojection loss and sparsity regularization:

$$L^{flow}_{rep} = L_{rep}\big(I_t,\ \mathrm{BW}(I_s,\ f^{full}_{t\to s})\big), \tag{6}$$

$$L^{flow}_{sparsity} = \big\|\omega(p_t)\, f^{res}_{t\to s}(p_t)\big\|_1, \tag{7}$$

where the spatial weight $\omega(p_t) = [\, L_{rep}(I_t, I_{s\to t})(p_t) < \bar{L}_{rep}\,]$, with $[\cdot]$ denoting the Iverson bracket, and $\bar{L}_{rep}$ representing the mean reprojection loss over all pixels.

Tri-Region Scene Decomposition. We then decompose dynamic scenes by jointly leveraging stereoscopic perception from depth prior D^{pr}_t and motion cues from residual flow $f^{res}_{t\to s}$. A coarse 3D structure is obtained by projecting image pixels into 3D space using D^{pr}_t. Specifically, for a pixel with homogeneous coordinates $\mathbf{u} = [u, v, 1]^\top$, its 3D position is computed as:

$$\mathbf{P}(u, v) = D^{pr}_t(u, v) \cdot K^{-1}\mathbf{u}. \tag{8}$$

Based on the reconstructed 3D points from the bottom half of the image, a ground plane M_g is estimated via RANSAC, following (Sun and Hariharan 2023).

To capture dynamic content, we first compute a motion-saliency map by summing bidirectional residual flows: $S = |f^{res}_{t\to t+1}| + |f^{res}_{t\to t-1}|$. We then binarize S at its global mean, producing an initial binary motion prior $M \in \{0, 1\}^{H\times W}$ that identifies pixels undergoing independent motion. Finally, We perform a connectivity analysis on M to obtain a set of candidate regions $\{\mathcal{R}\}^{N}_{i=1}$, each with a corresponding binary mask $\{M^{i}_{cand} \in \{0, 1\}^{H\times W}\}^{N}_{i=1}$.

- High-probability dynamic region ($\mathcal{R}_{hp\text{-}dyn}$, mask: $M_{hp\text{-}dyn}$). We define the high-probability dynamic region as the union of candidates whose 3D height above the estimated ground falls within the expected range of typical dynamic objects (e.g., pedestrians, vehicles). For each candidate region $\mathcal{R}^{i}_{cand}$ with mask M^{i}_{cand}, we compute the mean relative height h_i of its 3D points. Candidate regions satisfying

$$M_{hp\text{-}dyn} = \bigcup_{i=1}^{N} \Big\{ M^{i}_{cand} \,|\, \gamma \cdot h_{min} \le h_i \le \gamma \cdot h_{max} \Big\}, \tag{9}$$

are aggregated as the high-probability dynamic region. Here, $h_{min} = 0.3$ and $h_{max} = 2.5$ define the expected range of object height in metric space. The scale factor $\gamma = D^{pr}_{med}/30$ normalizes metric height by the predicted scene scale, with D^{pr}_{med} denoting the exponential moving average of the predicted median depth.

- Grounded uncertain region ($\mathcal{R}_{gu}$, mask: M_{gu}). Candidate regions that fall outside the expected height range may still correspond to dynamic context with appearance ambiguity (e.g., textureless or shadowed objects). To capture these cases, we impose a 2D spatial prior: a candidate region is considered potentially dynamic if it appears grounded in the image plane. Formally:

$$M_{gu} = \bigcup_{i=1}^{N} \Big\{ M^{i}_{cand} \,|\, \min_{y}(\mathbf{M}^{i}_{cand}) < \max_{y}(M_g) \Big\}, \tag{10}$$

where $\min_{y}(\cdot)$ and $\max_{y}(\cdot)$ denote the minimal and maximal y-coordinates of the respective masks. This rule selects the regions that visually contact the ground, despite uncertainty in their 3D geometry.

- High-probability static region ($\mathcal{R}_{hp\text{-}sta}$, mask: $M_{hp\text{-}sta}$). Pixels not covered by the high-confidence dynamic or grounded uncertain regions are assigned to a high-probability static region:

$$M_{hp\text{-}sta} = 1 - (M_{hp\text{-}dyn} \cup M_{gu})\,. \tag{11}$$

Region-Adaptive Depth Refinement Stage

The initial stage provides a coarse but geometrically consistent depth prior D^{pr}_t, along with a multi-region scene decomposition. However, the predicted depth remains inaccurate in dynamic pixels due to limited supervision signals and the accumulation of dynamic-related errors. To address this, we introduce a region-adaptive refinement stage that produces the final depth D_t by applying tailored supervision signals to different regions of the scene.

Motion-Robust Reprojection Loss. Photometric reprojection loss is the cornerstone of self-supervised monocular depth estimation, but it inherently assumes a static scene. Even after our multi-region decomposition, residual dynamic pixels in the grounded uncertain and high-probability static regions are still treated as static, which corrupts the photometric reprojection loss. To suppress this error propagation, we introduce a pseudo-static view synthesis that enforces a static scene assumption even in dynamic pixels.

Concretely, we employ a pseudo-rigid flow network Θ_{RIG} to predict the inverse pseudo-rigid flow $\tilde{f}^{rig}_{s\to t}$. The pseudo-static source image is obtained by differentiable backward warping:

$$\tilde{I}_s = BW\left(I_t, p_s + \tilde{f}^{rig}_{s\to t}(p_s)\right), \tag{12}$$

where p_s denotes homogeneous pixel coordinates. Using the final depth D_t and pose $T_{t\to s}$, we can reproject $\tilde{I}_s$ back into the target frame to obtain $\tilde{I}_{s\to t}$, as described in Eq. 1.

Our motion-robust reprojection loss is then applied outside the high-confidence dynamic mask $M_{hp\text{-}dyn}$:

$$L_{mrr} = (1 - M_{hp\text{-}dyn}) \cdot \min(L_{rep}(I_t, I_{s\to t}), L_{rep}(I_t, \tilde{I}_{s\to t})). \tag{13}$$

By selecting the lower of the two reprojection errors, the loss automatically favors the view (raw or pseudo-static) that best satisfies the static scene assumption, thus filtering out unreliable supervision signals.

To supervise the learning of pseudo-rigid flow predicted by Θ_{RIG}, we impose both rigid cycle consistency and photometric reprojection consistency within the high-probability static region $M_{hp\text{-}sta}$:

$$L_{rig} = M_{sp\text{-}sta} \cdot \left(\left\| f^{rig}_{t\to s} + \tilde{f}^{rig}_{s\to t}(\phi(p_t)) \right\|_1 + L_{rep}(I_s, \tilde{I}_s) \right), \tag{14}$$

where $\phi(p_t) = p_t + f^{rig}_{t\to s}(p_t)$ denotes the forward-warped pixel position in the source frame.

By integrating pseudo-static view synthesis into the reprojection loss, our method retains the expressive power of photometric supervision while robustly excluding dynamic outliers, resulting in more accurate and stable depth refinement in complex dynamic scenes.

Ground-Adherent Depth Constraint. To enhance depth estimation in dynamic objects without relying on semantic labels or external supervision, we propose a ground-adherent depth constraint (L_{gac}) that exploits structural cues between grounded objects and the ground. Our key insight is that many moving objects, such as vehicles and pedestrians, maintain a consistent spatial offset above the ground. Directly enforcing this relationship in regions of the high-probability dynamic region can be unstable, so we instead learn it from the more reliable grounded uncertain region. We then transfer this structural prior to the high-confidence dynamic region, using it to guide and stabilize depth predictions where motion introduces the greatest ambiguity.

As shown in Fig. 2 (b), we first identify contact contours from the estimated M_g by identifying boundary locations with near-vertical normals. These contours indicate object-ground contact and serve as geometric references. Within the uncertain ground region M_{gu} surrounding each contour, we sample anchor points j, at which we compute the local depth deviation:

$$\Delta_j = D_t(j) - D^{ground}_{med}(j), \tag{15}$$

where $D^{ground}_{med}(j)$ is the median depth of ground pixels in a small window around j. $\{\Delta_j\}$ are aggregated into a canonical offset δ. This offset captures the typical depth relationship between grounded objects and the ground.

To propagate this reference to pixels with more ambiguous motion, we apply the learned offset to the high-confidence dynamic region $M_{hp\text{-}dyn}$, constructing a reference depth for each anchor in $M_{hp\text{-}dyn}$ as:

$$D^{ref}_t(j) = D^{ground}_{med}(j) + \delta. \tag{16}$$

The ground-adherent depth constraint is defined as:

$$\begin{aligned} \mathcal{L}_{gac} = & \sum_{j\in M_{hp\text{-}dyn}} |D^{ref}_t(j) - D_t(j)| \\ & + M_{hp\text{-}dyn} \cdot (|\partial_y D^*_t| \, e^{-|\partial_y I_t|}). \end{aligned} \tag{17}$$

The ground-adherent depth constraint encourages geometric plausibility and depth coherence within dynamic regions. It combines a region-transferred depth prior to anchor object depths relative to the ground, and an edge-aware smoothness term to propagate depth consistently within each object.

Final Objective for Depth Refinement. The overall loss for the training of Θ_{FD} and Θ^F_{POSE} is:

$$L = \lambda_1 \cdot L_{mrr} + \lambda_2 \cdot (1 - M_{hp\text{-}dyn}) \cdot L_{rep} + \lambda_3 \cdot L_{gca} + \lambda_{sm} \cdot L_{sm}, \tag{18}$$

with $\lambda_1, \lambda_2, \lambda_3, \lambda_{sm}$ set to 0.85, 0.15, 0.05, and 0.001.

Experiments

Datasets

We train and evaluate our method on three widely adopted outdoor datasets: KITTI (Geiger, Lenz, and Urtasun 2012), Waymo Open (Sun et al. 2020), and Cityscapes (Cordts et al. 2016). For KITTI, we follow the Eigen split (Eigen and Fergus 2015) with 39,810 training images and 697 test images. For Cityscapes and Waymo, we adopt the splits from (Watson et al. 2021) and (Sun and Hariharan 2023), 69,731 and 76,852 training images, and 1,525 and 2,216 test images, respectively. Notably, KITTI contains predominantly static scenes, with only 0.6% of pixels belonging to dynamic objects (Nguyen et al. 2024), while Cityscapes and Waymo Open contain a significantly higher proportion of moving objects, posing greater challenges for self-supervised learning. To evaluate performance in dynamic regions, we use dynamic-class object masks from (Feng et al. 2022) for Cityscapes and static/dynamic masks from (Sun and Hariharan 2023) for Waymo Open. For depth evaluation metrics, the standard metrics (Eigen, Puhrsch, and Fergus 2014) are adopted, including error metrics (Abs Rel, Sq Rel, RMSE, and RMSE log) and accuracy metrics ($\delta < 1.25$, $\delta < 1.25^2$, and $\delta < 1.25^3$).

Implementation Details

For our pose networks Θ^P_{POSE} and Θ^F_{POSE}, we adopt the architecture from (Godard et al. 2019). Both Θ_{RES} and Θ_{RIG} follow the architecture of (Ilg et al. 2017). Our framework is integrated with several depth network architectures as the final depth network Θ_{FD}, including Monodepth2 (Godard et al. 2019), DiffNet (Zhou, Greenwood, and Taylor 2021), and LiteMono-8M (Zhang et al. 2023). The depth prior network Θ_{PD}, based on the lightweight LiteMono (Zhang et al. 2023), along with Θ^P_{POSE} and Θ_{RES}, are trained for 10 epochs in the first stage. In the second stage, Θ_{RIG} is optimized for 5 epochs, followed by a 20-epoch joint optimization of Θ_{PD} and Θ^F_{POSE}. The entire training needs 35 epochs, with an initial learning rate of 8e-5, which drops to 8e-6 after 30 epochs. The overall pipeline is trained on two NVIDIA RTX 4090D GPUs using Adam (Kingma 2014) optimizer with a batch size of 8.

	Method	OM	PM	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
KITTI	Li et al. (2021)	✓	-	0.130	0.950	5.138	0.209	0.843	0.948	0.978
	RM-Depth	✓	-	0.108	**0.710**	4.513	0.183	0.884	0.964	0.983
	Dynamo-Depth(MD2)	✓	-	0.120	0.864	4.850	0.195	0.858	0.956	0.982
	Dynamo-Depth(LiteMono-8M)	✓	-	0.112	0.758	4.505	0.183	0.873	0.959	**0.984**
	Monodepth2	-	-	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	Ours-Monodepth2	-	-	0.113	0.852	4.779	0.189	0.879	0.961	0.982
	DiffNet	-	-	0.102	0.764	4.483	0.180	0.896	0.965	0.983
	Ours-DiffNet	-	-	0.103	0.779	4.510	0.180	0.891	0.965	0.983
	LiteMono-8M	-	-	**0.101**	0.729	**4.454**	**0.178**	**0.897**	**0.965**	0.983
	Ours-LiteMono-8M	-	-	0.104	0.749	4.491	**0.178**	0.891	**0.965**	0.983
Waymo Open	Dynamo-Depth(MD2)	✓	-	0.130	1.439	6.646	0.183	0.851	0.959	0.985
	Dynamo-Depth(LiteMono-8M)	✓	-	0.116	1.156	6.000	**0.166**	0.878	**0.969**	**0.989**
	Struct2Depth	-	m	0.180	1.782	8.583	0.244	-	-	-
	Li et al. (2021)	-	m	0.157	1.531	7.090	0.205	-	-	-
	Lee et al. (2021b)	-	m	0.148	1.686	7.420	0.210	-	-	-
	Monodepth2†	-	-	0.173	2.731	7.708	0.227	0.797	0.930	0.968
	Ours-Monodepth2	-	-	0.130	1.531	6.403	0.180	0.869	0.964	0.985
	DiffNet †	-	-	0.158	2.414	7.432	0.211	0.817	0.944	0.976
	Ours-DiffNet	-	-	0.114	**1.123**	**5.990**	**0.166**	0.883	**0.969**	**0.989**
	LiteMono-8M†	-	-	0.158	2.305	7.394	0.210	0.816	0.944	0.976
	Ours-LiteMono-8M	-	-	**0.112**	1.145	6.025	**0.166**	**0.885**	**0.969**	**0.989**
Cityscapes	Li et al. (2021)	✓	-	0.119	1.290	6.980	0.190	0.846	0.952	0.982
	RM-Depth	✓	-	0.100	0.839	5.774	0.154	0.895	0.976	**0.993**
	DynamicDepth	-	m	0.103	1.000	5.867	0.157	0.895	0.974	0.991
	Mono-consistent-depth(DiffNet)	-	t	**0.085**	0.753	5.435	0.140	0.916	0.979	**0.993**
	From-Ground-To-Objects(MD2)	-	m	0.110	1.179	6.390	0.169	0.881	0.968	0.989
	From-Ground-To-Objects(MonoViT)	-	m	0.096	0.930	5.806	0.152	0.905	0.976	0.992
	Monodepth2†	-	-	0.129	1.569	6.876	0.187	0.849	0.957	0.983
	Ours-Monodepth2	-	-	0.097	0.940	5.743	0.149	0.903	0.975	0.992
	DiffNet†	-	-	0.140	3.345	7.767	0.215	0.839	0.940	0.971
	Ours-DiffNet	-	-	0.088	0.781	5.317	0.140	0.916	**0.980**	**0.993**
	LiteMono-8M†	-	-	0.150	3.315	7.831	0.228	0.826	0.930	0.964
	Ours-LiteMono-8M	-	-	**0.085**	**0.736**	**5.213**	**0.138**	**0.918**	**0.980**	**0.993**

Table 1: Performance comparison on the KITTI, Waymo Open, and Cityscapes datasets at resolutions of 640×192, 416×128, and 480×320, respectively. 'OM' stands for object motion modeling via 2D/3D flow. 'PM' indicates the use of pre-computed masks, where 'm' indicates pre-trained segmentation models and 't' indicates task-specific models for object identification. The best and the second best results are highlighted in **bold** and underlined, respectively. Manual replication with released code is indicated by †.

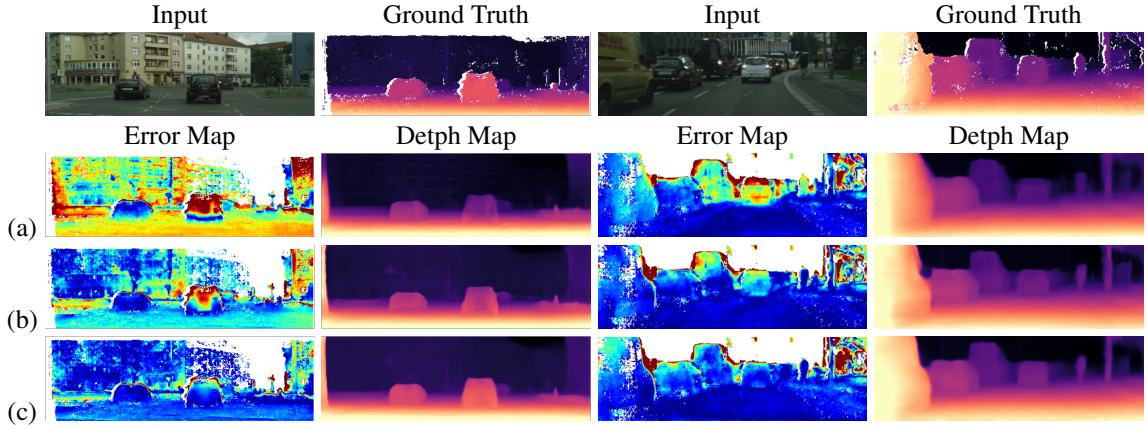

Figure 3: Examples of estimated depth maps and error maps in Cityscapes (Cordts et al. 2016). (a), (b), and (c) indicate the results of DynamicDepth (Feng et al. 2022), D_t^{pr}, and Ours-LiteMono-8M, respectively. In the error maps, blue indicates small errors, and red indicates large errors.

Results

Performance Comprison. Table 1 shows the performance comparison of our proposed framework and prior methods on the KITTI, Waymo Open, and Cityscapes datasets. Our method demonstrates significant improvements, particularly on the Waymo Open and Cityscapes datasets, which contain substantial dynamic object content. By integrating our approach, several existing depth networks, including Monodepth2 (Godard et al. 2019), DiffNet (Zhou, Green-

Method	PM	Abs Rel↓	Sq Rel↓	RMSE↓	$\delta < 1.25 \uparrow$
DynamicDepth (Feng et al. 2022)	m	0.129	1.273	4.626	0.862
From-Ground-To-Objects(MD2)	m	0.136	1.238	4.791	0.864
From-Ground-To-Objects(MonoViT)	m	0.109	0.888	4.243	0.898
Monodepth2		0.159	1.937	6.363	0.816
Ours-Monodepth2		0.096	0.784	3.942	0.902
DiffNet		0.185	2.751	6.219	0.793
Ours-DiffNet		0.085	0.615	3.628	0.920
LiteMono-8M		0.187	2.771	6.348	0.776
Ours-LiteMono-8M		**0.084**	**0.609**	**3.523**	**0.924**

Table 2: Performance comparison for objects in dynamic classes on Cityscapes.

Method	Moving Object		Static Background	
	Abs Rel↓	$\delta < 1.25 \uparrow$	Abs Rel↓	$\delta < 1.25 \uparrow$
Dynamo-Depth(MD2)	0.234	0.674	0.122	0.851
Dynamo-Depth	0.194	0.750	0.110	0.891
Monodepth2	0.749	0.416	0.152	0.810
Ours-Monodepth2	0.229	0.723	0.120	0.872
DiffNet	0.613	0.505	0.139	0.831
Ours-DiffNet	0.195	0.780	**0.102**	0.906
LiteMono-8M	0.599	0.506	0.140	0.827
Ours-LiteMono-8M	**0.191**	**0.784**	**0.102**	**0.909**

Table 3: Performance comparison for moving objects and static background on Waymo Open.

wood, and Taylor 2021), and LiteMono-8M (Zhang et al. 2023) achieve notable performance gains. On Cityscapes, these networks exhibit improvements of 24.8%, 37.1%, and 43.4% in the Abs Rel metric, respectively. Notably, Ours-LiteMono-8M outperforms all previous methods by significant margins across all metrics.

Our method, relying solely on the input images, outperforms approaches that leverage pre-trained segmentation models (e.g., DynamicDepth (Feng et al. 2022), From-Ground-To-Objects (Moon et al. 2024)) or multi-task-specific models with complex training processes (e.g., Mono-consistent-depth (Nguyen et al. 2024)). Table 2 evaluates the performance on dynamic-class objects regions on Cityscapes, following the process detailed in (Feng et al. 2022; Moon et al. 2024). Our models significantly outperform existing methods (Feng et al. 2022; Moon et al. 2024) in all metrics. Fig. 2 provides qualitative comparisons on the Cityscapes test set. More qualitative results are included in the supplementary material.

On the challenging Waymo Open dataset, our method demonstrates strong generalization, consistently improving baseline models across all metrics (Table 1). Unlike Dynamo-Depth (Sun and Hariharan 2023), which models independent motion for dynamic objects with insufficient supervision, our approach leverages motion cues to identify dynamic regions and provide robust supervision. This leads to significant improvements in both moving objects and static backgrounds, as shown in Table 3.

As shown in Table 1, our models perform competitively with state-of-the-art methods on the KITTI dataset. However, due to the limited presence of dynamic objects in KITTI (Sun and Hariharan 2023; Nguyen et al. 2024), the advantages of explicitly handling dynamic regions are less evident. Notably, by employing the same depth network as Dynamo-Depth (Sun and Hariharan 2023) but avoiding explicit modeling of independent motion, our method reduces the inherent ambiguity associated with such motion and achieves superior performance. Furthermore, this design also alleviates uncertainty in static regions, resulting in additional performance gains on KITTI.

GMoDecomp	L_{mrr}	L_{gac}	Dynamic-class Region		All Region	
			Abs Rel ↓	$\delta < 1.25 \uparrow$	Abs Rel ↓	$\delta < 1.25 \uparrow$
-	-	-	0.187	0.776	0.150	0.826
✓			0.098	0.901	0.096	0.905
✓	✓		0.089	0.915	0.089	0.914
✓	✓	✓	**0.084**	**0.924**	**0.085**	**0.918**

Table 4: Ablation study of our method using LiteMono-8M (Zhang et al. 2023) as Θ_{FD} on Cityscapes (Cordts et al. 2016). Training with GMoDecomp only indicates the performance from depth prior D_t^{pr} in the first stage.

Ablation Study

We conduct the ablation study to assess the effectiveness of our methods by incrementally applying our contributions and evaluating performance on the Cityscapes (Cordts et al. 2016) dataset, as shown in Table 4.

GMoDecomp. Our proposed GMoDecomp performs dynamic scene decomposition in the initial structural inference stage and ensures the robust depth prior D_t^{pr} generation. As shown in Table 4, D_t^{pr} with the help of decomposition results from GMoDecomp shows marginal improvement in both objects in dynamic classes and all regions.

$\boldsymbol{L_{mrr}}$**.** Leveraging GMoDecomp, we can identify regions dominated by static pixels. However, directly applying L_{rep} in Eq. 5 in these regions propagates errors due to misclassified dynamic pixels. As shown in Table 4, incorporating L_{mrr} in Eq. 13 not only improves overall performance but also enhances depth accuracy in dynamic object regions.

$\boldsymbol{L_{gac}}$**.** To provide reliable supervision for dynamic objects, L_{gac} in Eq. 17 leverages the relationship between the depth of static objects and the ground contacting them. As shown in Table 4, L_{gac} significantly improves depth estimation accuracy in dynamic object regions. Additional ablation studies on auxiliary loss terms and hyperparameter settings are provided in the supplementary material.

Conclusion

We introduce AdaDepth, a self-supervised framework designed to improve monocular depth estimation in dynamic scenes. The proposed geometry-motion joint scene decomposition (GMoDecomp) module jointly exploits structural priors and motion cues to perform tri-region 3D-aware scene decomposition, enabling the generation of a robust and geometrically consistent depth prior. In the region-adaptive refinement stage, we leverage the depth prior and scene decomposition to selectively propagate supervision, mitigating depth distortion in dynamic objects. The extensive experimental results show that our method is very well harmonized with existing depth estimation methods to effectively handle such moving objects.

Acknowledgments

This work is partially supported by NSFC (No. 62376153, 62402318, 24Z990200676, 62272419, BC0301580).

References

Bian, J.; Li, Z.; Wang, N.; Zhan, H.; Shen, C.; Cheng, M.-M.; and Reid, I. 2019. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems*, 32.

Casser, V.; Pirk, S.; Mahjourian, R.; and Angelova, A. 2019. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 8001–8008.

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

Eigen, D.; and Fergus, R. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, 2650–2658.

Eigen, D.; Puhrsch, C.; and Fergus, R. 2014. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27.

Feng, Z.; Yang, L.; Jing, L.; Wang, H.; Tian, Y.; and Li, B. 2022. Disentangling object motion and occlusion for unsupervised multi-frame monocular depth. In *European Conference on Computer Vision*, 228–244. Springer.

Garg, R.; Bg, V. K.; Carneiro, G.; and Reid, I. 2016. 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*, 740–756. Springer.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.

Godard, C.; Mac Aodha, O.; and Brostow, G. J. 2017. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 270–279.

Godard, C.; Mac Aodha, O.; Firman, M.; and Brostow, G. J. 2019. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3828–3838.

Guizilini, V.; Ambrus, R.; Pillai, S.; Raventos, A.; and Gaidon, A. 2020a. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2485–2494.

Guizilini, V.; Ambruș, R.; Chen, D.; Zakharov, S.; and Gaidon, A. 2022. Multi-frame self-supervised depth with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 160–170.

Guizilini, V.; Hou, R.; Li, J.; Ambrus, R.; and Gaidon, A. 2020b. Semantically-guided representation learning for self-supervised monocular depth. *arXiv preprint arXiv:2002.12319*.

Han, W.; Yin, J.; Jin, X.; Dai, X.; and Shen, J. 2022. Brnet: Exploring comprehensive features for monocular depth estimation. In *European Conference on Computer Vision*, 586–602. Springer.

Han, W.; Yin, J.; and Shen, J. 2023. Self-Supervised Monocular Depth Estimation by Direction-aware Cumulative Convolution Network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8613–8623.

Hui, T.-W. 2022. Rm-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1675–1684.

Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2462–2470.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. *Advances in neural information processing systems*, 28.

Jung, H.; Park, E.; and Yoo, S. 2021. Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12642–12652.

Kingma, D. P. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klingner, M.; Termöhlen, J.-A.; Mikolajczyk, J.; and Fingscheidt, T. 2020. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, 582–600. Springer.

Lee, S.; Im, S.; Lin, S.; and Kweon, I. S. 2021a. Learning monocular depth in dynamic scenes via instance-aware projection consistency. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 1863–1872.

Lee, S.; Rameau, F.; Pan, F.; and Kweon, I. S. 2021b. Attentive and contrastive learning for joint depth and motion field estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4862–4871.

Li, H.; Gordon, A.; Zhao, H.; Casser, V.; and Angelova, A. 2021. Unsupervised monocular depth learning in dynamic scenes. In *Conference on Robot Learning*, 1908–1917. PMLR.

Luo, C.; Yang, Z.; Wang, P.; Wang, Y.; Xu, W.; Nevatia, R.; and Yuille, A. 2019. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10): 2624–2641.

Lyu, X.; Liu, L.; Wang, M.; Kong, X.; Liu, L.; Liu, Y.; Chen, X.; and Yuan, Y. 2021. Hr-depth: High resolution

self-supervised monocular depth estimation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 2294–2301.

Mahjourian, R.; Wicke, M.; and Angelova, A. 2018. 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*, 5667–5675.

Moon, J.; Bello, J. L. G.; Kwon, B.; and Kim, M. 2024. From-Ground-To-Objects: Coarse-to-Fine Self-supervised Monocular Depth Estimation of Dynamic Objects with Ground Contact Prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10519–10529.

Nguyen, H. C.; Wang, T.; Alvarez, J. M.; and Liu, M. 2024. Mining Supervision for Dynamic Regions in Self-Supervised Monocular Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10446–10455.

Ranjan, A.; Jampani, V.; Balles, L.; Kim, K.; Sun, D.; Wulff, J.; and Black, M. J. 2019. 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*, 12240–12249.

Shu, C.; Yu, K.; Duan, Z.; and Yang, K. 2020. Feature-metric loss for self-supervised learning of depth and ego-motion. In *European Conference on Computer Vision*, 572–588. Springer.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2446–2454.

Sun, Y.; and Hariharan, B. 2023. Dynamo-depth: Fixing unsupervised depth estimation for dynamical scenes. *Advances in Neural Information Processing Systems*, 36: 54987–55005.

Vankadari, M.; Golodetz, S.; Garg, S.; Shin, S.; Markham, A.; and Trigoni, N. 2023. When the sun goes down: Repairing photometric losses for all-day depth estimation. In *Conference on Robot Learning*, 1992–2003. PMLR.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Watson, J.; Mac Aodha, O.; Prisacariu, V.; Brostow, G.; and Firman, M. 2021. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1164–1174.

Woo, S.; Lee, W.; Kim, W. J.; Lee, D.; and Lee, S. 2025. ProDepth: Boosting Self-Supervised Multi-Frame Monocular Depth with Probabilistic Fusion. In *European Conference on Computer Vision*, 201–217. Springer.

Yin, Z.; and Shi, J. 2018. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1983–1992.

Zhang, N.; Nex, F.; Vosselman, G.; and Kerle, N. 2023. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18537–18546.

Zhou, H.; Greenwood, D.; and Taylor, S. 2021. Self-supervised monocular depth estimation with internal feature fusion. *arXiv preprint arXiv:2110.09482*.

Zhou, T.; Brown, M.; Snavely, N.; and Lowe, D. G. 2017. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1851–1858.

Zou, Y.; Luo, Z.; and Huang, J.-B. 2018. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European conference on computer vision (ECCV)*, 36–53.