# Towards Robust Driving Perception: A Flexible Scale-Driven Family for Self-Supervised Monocular Depth Estimation

Zhaowen Zhu[1,3*], Li Zhang[1†], Yujie Chen[1*], Tian Zhang[1,4], Yingjie Wang[2], Mingxia Zhan[1]

[1] Hefei University of Technology, Hefei, China
[2] Nanjing University of Posts and Telecommunications, Nanjing, China
[3] Chengpin Home Tech, Nanjing, China
[4] Differential Robotics, Hangzhou, China

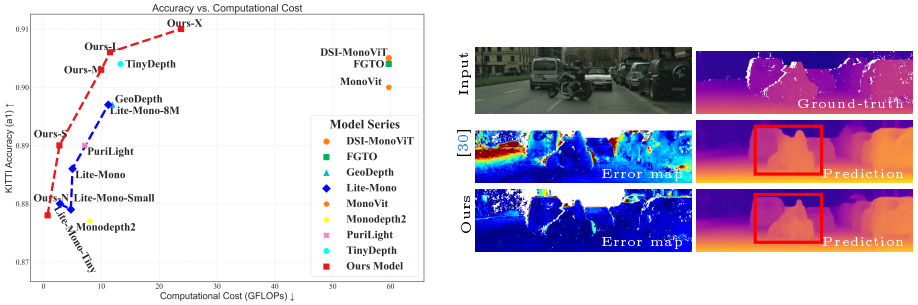


**Fig. 1: Performance and visual comparison.** (left) Accuracy vs. cost on KITTI [12]. (right) Cityscapes [6]: sharper boundaries and robust dynamic depth.

**Abstract.** Self-Supervised Monocular Depth Estimation (MDE) has garnered attention in recent years due to its independence from ground truth. However, most existing models are limited to a single scale and exhibit considerable performance degradation in complex driving environments. Networks specifically designed to handle dynamic traffic participants tend to be overly complex, hindering their deployment on resource-constrained automotive edge devices. To address these limitations and move towards robust driving perception, we propose FlexDepth, a scale-driven and flexible family of self-supervised MDE models tailored for challenging road scenarios. FlexDepth employs a two-stage static-dynamic decoupled training strategy, enabling the independent assessment of confidence for both static backgrounds and dynamic road objects. Furthermore, it introduces a meticulously designed Scale-Driven Decoder (SDD) to dynamically select components based on scale size, facilitating efficient feature fusion and the output of high-precision depth maps. Extensive experiments on standard driving benchmarks demonstrate that without any auxiliary information, our model achieves state-of-the-art

* Equal contribution.
† Corresponding author: lizhang@hfut.edu.cn

performance across arbitrary scales with minimal computational overhead. Our smallest model, Flex-Nano, requires only 0.7 GFLOPs and achieves 37.6 FPS on mobile platforms, ensuring reliable real-time perception while maintaining excellent zero-shot generalization. Our source code is avalible: https://github.com/startnew/flexdepth

**Keywords:** Monocular Depth Estimation · Dynamic environments · Scale-Driven

## 1 Introduction

Monocular depth estimation aims to recover 3D scene geometry from a single image and plays a fundamental role in autonomous driving, robotic navigation, and Unmanned Aerial Vehicle (UAV) perception. Due to the scarcity of dense ground-truth depth in real-world scenarios, self-supervised approaches [11,14,16, 42, 51] have gained significant attention. These methods construct supervision from stereo pairs [11, 13, 15] or monocular video sequences [14, 42, 55], enabling scalable learning without explicit depth labels. Early models often rely on single-scale designs [1, 20, 39] and degrade notably in dynamic scenes [57]. Later works improve global reasoning by adopting Vision Transformers [33, 55] or diffusion priors [17, 21, 34], but at the cost of substantially increased model complexity and inference latency.

To reduce computational overhead, lightweight architectures have been explored [18, 28, 51, 55, 57]. Zhang *et al.* [51] combined CNN and Transformer modules, while Chen *et al.* [4] introduced a lightweight shuffle and purification framework. However, these designs often compromise structural fidelity, particularly around thin structures and object boundaries, and still struggle in dynamic environments. Since most self-supervised methods rely on photometric reprojection, moving objects violate static-scene assumptions and degrade supervision quality. Existing attempts include motion disentanglement [2, 10, 42, 43], semantic or flow assistance [18, 23, 25, 30, 37], and cross-dataset or pseudo-label training strategies [27, 49, 53]. Although Nimbled [27] enhances supervision via pseudo-label refinement and large-scale video pretraining, these approaches typically introduce additional priors, auxiliary tasks, or complex pipelines, limiting flexibility and deployment efficiency.

Parallelly, supervised foundation models like Metric3D [48] and Depth Anything V2 [45] have set new benchmarks in zero-shot generalization. However, their reliance on massive curated datasets and heavy backbones limits their deployment in resource-constrained environments. In contrast, self-supervised frameworks offer superior flexibility and environmental adaptability. Nevertheless, existing lightweight self-supervised designs struggle to maintain structural fidelity, particularly near object boundaries, and exhibit poor robustness in dynamic environments such as driving scenes where moving objects violate the static-camera assumption. Current solutions for handling dynamic scenes typically introduce auxiliary tasks or complex pipelines, which limit their deployment capabilities on edge devices. Thus, improving robustness and scalability within

lightweight self-supervised frameworks remains an important complementary direction.

Consequently, a key question arises: *can self-supervised monocular depth estimation narrow the robustness gap with large supervised models while preserving lightweight design and data efficiency?* Addressing this requires jointly reconsidering architectural scaling and dynamic-scene handling within a unified optimization framework rather than relying on larger backbones or external supervision.

To this end, we propose FlexDepth, a family of scale-driven self-supervised monocular depth models that balance architectural efficiency with robustness in driving scenarios. Unlike traditional lightweight decoders that rely on simple upsampling methods, we propose a Scale-Driven Decoder (SDD). The SDD adaptively selects its constituent components to enhance feature fusion and yield high-precision depth maps. Furthermore, this paper introduces a two-stage static-dynamic decoupled training strategy to disentangle dynamic driving environments from static backgrounds. By performing separate confidence assessments, our approach achieves superior robustness in highly dynamic environments. Fig. 1 illustrates the core contributions of our work. The FlexDepth model family achieves state-of-the-art performance on the KITTI [12] dataset with low computational overhead and reduced inference latency.

Our contributions are summarized as follows:

- We propose FlexDepth, a scale-driven family of lightweight self-supervised depth models that systematically study accuracy–efficiency trade-offs while maintaining robustness in dynamic driving environments.
- We design a two-stage static-dynamic decoupled training strategy that enables independent confidence assessments for static backgrounds and dynamic objects, thereby providing more effective guidance for the training of the depth prediction network.
- A novel Scale-Driven Decoder (SDD) is introduced to facilitate efficient inference and precise depth mapping, supporting scalable deployment on edge devices according to specific scene requirements.

## 2 Related Work

### 2.1 Self-Supervised Monocular Depth Estimation

Owing to the limited availability of high-quality ground-truth depth data, self-supervised monocular depth estimation has attracted considerable research interest. Garg *et al.* [11] first formulated the task as a novel view synthesis problem using stereo imagery. Subsequently, a series of studies have been proposed, such as designing robust loss functions [36, 50, 54], incorporating auxiliary information during training [22, 41], and introducing additional geometric constraints [24, 47]. Godard *et al.* [14] further advanced the field with Monodepth2, which introduced a per-pixel minimum reprojection loss to handle occlusions and a multi-scale sampling strategy to recover finer details. Recently, Zhang et al. [53] proposed Hybrid-depth, a framework that leverages large-scale multimodal backbones (CLIP [32] and DINO [3, 31]) .

### 2.2 Efficient Architecture Design

Building upon the advancements in self-supervised monocular depth estimation, a growing research direction has focused on developing lightweight models [5, 40, 51, 57] to overcome the computational bottlenecks associated with deploying these systems on resource-constrained edge devices. Lyu *et al.* [28] adopted MobileNetV3 and a Feature Fusion Squeeze-Excitation module, while Zhou *et al.* [57] introduced R-MSFM with a parameter-sharing decoder to cut training parameters. Zhang *et al.* [51] combined CNN and Transformer using dilated convolution and cross-channel covariance attention. Chen *et al.* [4] implemented global feature purification by operating in the frequency domain. However, these models often sacrifice structural accuracy or incur high computational costs, failing to balance performance and lightweight design. Moreover, they struggle in dynamic environments, thereby limiting their robustness in autonomous driving scenarios.

### 2.3 Adaptation to Dynamic Environment

Constrained by reprojection loss, mitigating the impact of dynamic objects on model training has become a challenge in self-supervised approaches. To address this issue, Monodepth2 [14] introduced an auto-masking mechanism, which, however, is only effective for objects moving at the same velocity as the observer. Other methods integrate depth estimation with semantic segmentation [10, 29] or optical flow [18, 23, 25, 30, 37], but they struggle in unfamiliar scenes and cannot eliminate side effects like shadows or motion blur. Recently, Zhang *et al.* [52] proposed the DSI framework to address these issues, yet its reliance on empirical hyperparameters restricts generalization. Furthermore, most of these approaches rely heavily on computationally expensive backbones and external priors, which hinders their efficient deployment on resource-constrained edge devices.

## 3 Method

This paper introduces a scale-driven family of flexible self-supervised monocular depth estimation models, termed FlexDepth, designed for dynamic driving environments. Fig. 2 shows the proposed architecture. The FlexDepth model family comprises five models across varying scales—Nano, Small, Medium, Large, and X-Large—tailored for flexible edge-side deployment based on specific resource constraints. Central to this architecture is our Scale-Driven Decoder (SDD), which adaptively selects specific modules and training pipelines corresponding to each model scale. Furthermore, the training process is structured into two distinct stages: the first stage optimizes the pose and depth networks to establish a lightweight yet efficient framework. In the second stage, a simple yet effective static-dynamic decoupled mask is introduced to facilitate the retraining of the depth network. This decoupling enhances the model's representational power in dynamic driving scenes, ultimately yielding superior robustness and environmental adaptability without compromising efficiency.

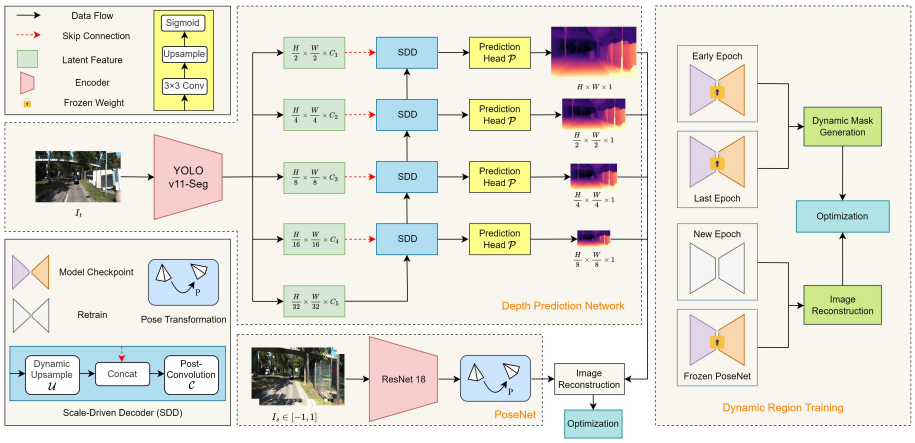


**Fig. 2: FlexDepth system overview.** FlexDepth implements a two-stage training pipeline. In the first stage, both the Depth Prediction Network and the PoseNet are optimized simultaneously. The second stage focuses on dynamic scene refinement, where motion-induced epoch-wise inconsistency is leveraged to generate dynamic object masks. Subsequently, the Depth Prediction Network is optimized using the frozen PoseNet.

### 3.1 Motivation and Analysis

In the field of lightweight self-supervised monocular depth estimation, research has primarily focused on encoder design [4, 14, 51], while decoders are often marginalized with simplified feature fusion and upsampling methods. Although encoders can extract rich representations, the inductive bias inherent in U-Net-like architectures acts as a bottleneck, limiting the transformation of low-resolution abstract features into precise depth maps. We argue that the decoder is equally crucial for overall performance gain. To this end, we revisit decoder construction and propose the Scale-Driven Decoder (SDD), which dynamically selects model components and training strategies based on varying model scales. Furthermore, from the perspectives of upsampling and feature integration, we demonstrate that models of different scales exhibit distinct dependencies. This highlights the rationality and superiority of SDD compared to single-scale, fixed-component architectures. Our approach ultimately achieves real-time inference of high-precision depth maps.

The processing of dynamic scenes has emerged as a prominent research area in recent years. However, most existing models [29, 30, 37, 43] for handling dynamic environments exhibit high computational complexity or require extensive prior knowledge, posing significant challenges for deployment in resource-constrained settings. By adopting a two-stage training approach complemented by a simple yet efficient static-dynamic decoupled mask, our method evaluates confidence levels for static and dynamic regions separately, enabling precise distinction between them.

### 3.2 Depth Prediction Network

**Depth Encoder.** We adopt a standard multi-scale feature extraction architecture based on YOLOv11 [19] as the depth encoder. Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, the encoder progressively generates five feature maps through successive downsampling operations, with spatial resolutions ranging from $\frac{1}{2}$ to $\frac{1}{32}$ of the original input size.

**Depth Decoder.** Conventional approaches [4,14,51] typically follow a pipeline of "pre-convolution → upsampling → concatenation → post-convolution". Most lightweight tasks often prioritize this pipeline, which applies convolution kernels prematurely before scale restoration to reduce computational overhead. However, this forces the compression of already sparse global semantic information, leading to the irreversible loss of critical depth cues before spatial expansion occurs. This typically manifests as blurred object boundaries in depth maps and a failure to predict small-scale objects. We rethink this fixed pipeline in lightweight networks, maintaining that the pursuit of efficiency should not come at the cost of global semantic information in deep features. Consequently, we remove the pre-convolution stage entirely, shifting our design focus toward upsampling methods $\mathcal{U}$, post-convolution $\mathcal{C}$, and the prediction head $\mathcal{P}$. The depth map $D$ generation process in our revised pipeline can be formulated as follows:

$$D_i = \mathcal{P}\{\mathcal{C}(\text{Concat}[\mathcal{U}(\mathbf{F}_{i+1}), \mathbf{F}_{enc,i}])\}, \tag{1}$$

where $\mathbf{F}_{i+1}$ is the $(i+1)$-th layer decoder feature, and $\mathbf{F}_{enc,i}$ is the $i$-th layer encoder. They are concatenated via a skip connection. The implementation of the FlexDepth model across different scales varies specifically in terms of $\mathcal{C}$, $\mathcal{U}$, and $\mathcal{P}$. We begin by introducing these three key components.

**Post-Convolution $\mathcal{C}$.** Since the pre-convolution module has been removed, the design of the post-convolution module becomes particularly crucial. Drawing inspiration from [19], we designed a combination of two types of Bottleneck units (B): the High-Efficiency Bottleneck (HEB) and the High-Performance Bottleneck (HPB). The output of each HEB layer is concatenated into the final output. This dense connection shortens the path for gradient backpropagation, making the model easier to train. HEB is particularly suitable for small scale models that prioritize gradient flow richness and stability. Given the input feature $\mathbf{X}$ and the uniform split operator $\mathcal{S}$, the output $\mathbf{X}_{out}$ can be represented as:

$$[f_a, f_b] = \mathcal{S}(Conv(X)), \tag{2}$$

$$F_1 = B_1(f_a), F_2 = B_2(F_1), \cdots, F_n = B_n(F_{n-1}), \tag{3}$$

$$X_{out} = Conv(Concat[f_a, f_b, F_1, \cdots, F_n]). \tag{4}$$

On the other hand, the core logic of HPB is similar to that of HEB. The primary distinction lies in the replacement of the standard Bottleneck unit $B$ with a bottleneck block $B^k$ featuring a Cross Stage Partial (CSP) structure. Furthermore, the feature concatenation logic places greater emphasis on modular CSP

fusion. This approach offers the advantage of capturing finer-grained local features, thereby enhancing model performance without significantly increasing the computational overhead. The process of $B^k$ is illustrated as follows:

$$B^k(x) = Conv(Concat[\prod_{j=1}^{n} B_j^{'}(Conv(x)), Conv(x)]), \tag{5}$$

where $\prod$ denotes the sequential composition of bottleneck layers. To distinguish from the outer block $B_i$, $B_j'$ represents the minimal convolutional unit within the internal CSP structure.

**Upsampling Operation $\mathcal{U}$.** Inspired by [26], we develop a dynamic upsampling approach tailored for models across various scales. Traditional lightweight upsampling methods based on simple bilinear interpolation are inherently content-agnostic; they apply fixed upsampling weights to every region of an image, regardless of whether it is a smooth background or a complex edge, which frequently results in blurred boundaries. Furthermore, as a form of simple weighted averaging, bilinear interpolation exerts a low-pass filtering effect that suppresses high-frequency information, leading to halo artifacts or blurring at object contours.To address these limitations, we implement a two-stage process for each scale: offset generation and dynamic resampling. By leveraging a convolutional layer to predict upsampling offsets $\Delta P$ dynamically, the model can adaptively determine the optimal sampling locations from the original feature map based on semantic content, such as object silhouettes. By integrating a sampling grid with these predicted offsets, we achieve non-uniform upsampling, thereby generating sharper and more accurate structural boundaries. The general procedure for the dynamic upsampling operation $\mathcal{U}$ is formulated as:

$$\mathcal{U}(X) = GS(X, PS(\frac{2 \cdot (\mathbf{C} + \Delta\mathbf{P})}{\text{Norm}} - 1, s)), \tag{6}$$

where $\mathbf{C}$ denotes the uniform sampling grid coordinates, Norm represents the normalization factor, and $s$ is the upsampling scale factor. $PS$ refers to the Pixel Shuffle operator, which transforms the predicted offset information into the spatial coordinate dimension. $GS$ stands for the Grid Sample function, which performs bilinear interpolation based on the calculated coordinates.

It is worth noting that for larger models, we experimentally employ an inverted upsampling prediction head $\mathcal{P}$, which reverses the traditional sequence of prediction followed by upsampling. The rationale is that high-channel features maintain structural and semantic consistency after upsampling, thereby enabling subsequent prediction convolutions to generate high-quality outputs more accurately. Conversely, smaller models adhere to the traditional paradigm to preserve real-time inference speeds on resource-constrained devices.

In summary, FlexDepth (Nano/Small) utilize a Post-Convolution $\mathcal{C}$ based on HEB, the dynamic upsampling operator $\mathcal{U}$, and a traditional prediction head $\mathcal{P}$. Conversely, the Medium, Large, and X-Large versions adopt a Post-Convolution $\mathcal{C}$ based on HPB, the dynamic upsampling operator $\mathcal{U}$, and an inverted prediction head $\mathcal{P}$.

### 3.3 Two-Stage Static-Dynamic Decoupled Training

Recent research in dynamic environments has demonstrated that dynamic regions exhibit significantly higher predictive instability during the training process. Despite this observation, most existing methodologies rely heavily on complex network architectures and extensive prior knowledge, which inherently limits their potential for lightweight deployment. Inspired by the work of Zhang et al. [52], we propose a static-dynamic decoupling approach. A fundamental limitation of [52] lies in its reliance on a fixed threshold for region differentiation. In real-world applications, the proportion of dynamic objects varies significantly across different scenes and individual frames. Consequently, methodologies that depend on dataset-wide statistics lack the necessary robustness to generalize across diverse and unpredictable environments. We employ a two-stage training strategy to achieve robustness in dynamic scenes without introducing additional architectural complexity or requiring prior environmental knowledge. At its core, our method focuses on the retraining of the depth prediction network to better handle temporal inconsistencies. Specifically, the training instability of dynamic regions is characterized by the discrepancy between the model's outputs during the early and late stages of training. While the predicted depth of static backgrounds remains relatively consistent throughout the training process, dynamic regions exhibit significant fluctuations. To address this, we designed the Static-Dynamic Decoupled Mask to effectively decouple the static and dynamic components of the environment, thereby enhancing the model's overall stability.

In the initial stage, we simultaneously train the depth prediction network and the pose estimation network. However, due to the inherent predictive instability of moving objects, their depth outputs exhibit significant variance between early and late training epochs, whereas static backgrounds remain relatively consistent. To leverage this instability for decoupling, we compute disparity maps using weights from the initial ($disp_i$) and final ($disp_l$) epochs of the first stage. The core of the second-stage training involves constructing a static-dynamic decoupled mask $M$ to recalibrate the confidence assessments for dynamic objects and the static background:

$$M = [|\text{disp}_i^{\beta} - \text{disp}_l^{\beta}| < g(\phi(f_{s_i}, f_{s_l}))], \tag{7}$$

where $\beta$ is set to 0.7, $[\cdot]$ means the Iverson bracket. $g(\phi(f_{s_i}, f_{s_l}))$ serves as an adaptive thresholding function. Specifically, $\phi$ measures the structural evolution between early-stage ($f_{s_i}$) and late-stage ($f_{s_l}$) feature representations, while $g$ (See Appendix) maps these variances into a pixel-wise threshold space. This mechanism allows the network to dynamically adjust its sensitivity: in regions with high semantic complexity or rapid motion, the threshold adapts to ensure that only stable, static pixels contribute to the subsequent confidence evaluation. Our method essentially conducts a refined retraining of the depth network, achieving superior robustness in highly heterogeneous dynamic environments without any additional inference overhead.

### 3.4 Pose Network and Self-Supervised Training

Following the self-supervised paradigm [14, 56], we jointly optimize the depth and pose networks through photometric consistency without ground truth supervision. Concretely, our PoseNet adopts a ResNet-18 encoder with a four-layer decoder to predict the relative 6-DoF pose transformation $P_{t\to s}$ between adjacent frames.

To train the networks, we warp the source image $I_s$ into the target view to generate the reconstructed image $\hat{I}_t$ according to the predicted depth $\hat{D}_t$ and pose $P_{t\to s}$:

$$\hat{I}_t = I_s\langle \text{proj}(\hat{D}_t, P_{t\to s}, K)\rangle, \tag{8}$$

where $\langle\cdot\rangle$ denotes the differentiable bilinear sampling operator following [14]. The photometric error is formulated as:

$$L_p = \alpha\frac{1-\text{SSIM}(\hat{I}_t, I_t)}{2} + (1-\alpha)\|\hat{I}_t - I_t\|_1, \tag{9}$$

where $\alpha = 0.85$ by default. To handle occlusions and moving objects, we employ the minimum reprojection loss with auto-masking [14]:

$$L_r = \mu \cdot \min_s L_p(\hat{I}_t, I_t), \tag{10}$$

where $\mu$ excludes pixels with inconsistent photometric errors. We also adopt the edge-aware smoothness loss [13] to regularize the predicted depth:

$$L_{\text{smooth}} = |\partial_x d_t^*|e^{-|\partial_x I_t|} + |\partial_y d_t^*|e^{-|\partial_y I_t|}, \tag{11}$$

where $d_t^*$ denotes the mean-normalized inverse depth. The total loss for the first training stage is:

$$L_{raw} = \frac{1}{4}\sum_{s\in\{1,\frac{1}{2},\frac{1}{4},\frac{1}{8}\}}(L_r + \lambda L_{\text{smooth}}), \tag{12}$$

where $\lambda = 0.001$ by default.

In the second stage, with the static-dynamic decoupled mask $M$ obtained from Sec. 3.3, we incorporate the masked normal distribution loss [52]:

$$L_m = M \cdot \left[\log(\sigma+1) + \frac{\lambda L_r^2}{2\sigma^2}\right], \tag{13}$$

where $\lambda = 0.5$ and $\sigma$ is predicted by a variance network. We further apply geometric depth smoothing [52]:

$$L_{\text{GDS}} = |\partial_x \hat{d}_t|e^{-|\partial_x I_t|} + [\eta(1-M)+M]|\partial_y \hat{d}_t|e^{-|\partial_y I_t|}, \tag{14}$$

with $\eta = 100$. The final loss is formulated as $L_{\text{total}} = \frac{1}{HW}\sum_t(L_m + \gamma L_{\text{GDS}})$.

**Table 1:** Quantitative Results on the KITTI [12] (Eigen split [8]) and Cityscapes [6] datasets. The input resolutions are $640 \times 192$ for KITTI and $416 \times 128$ for Cityscapes. **M**: monocular videos, **M+Se**: monocular videos + semantic segmentation, **T.F.**: test input frames, **Bold** denotes the best result.

| D | Method | Year | T.F. | Data | Depth Error↓ | | | | Depth Accuracy↑ | | | FLOPs.(G)↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Abs Rel | Sq Rel | RMSE | RMSE log | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ | |
| KITTI | Monodepth2 [14] | 2019 | 1 | M | 0.115 | 0.903 | 4.863 | 0.193 | 0.877 | 0.959 | 0.981 | 8.0 |
| | Lite-Mono-Tiny [51] | 2023 | 1 | M | 0.110 | 0.837 | 4.710 | 0.187 | 0.880 | 0.960 | 0.982 | 2.8 |
| | Lite-Mono-Small [51] | 2023 | 1 | M | 0.110 | 0.802 | 4.671 | 0.186 | 0.879 | 0.961 | 0.982 | 4.7 |
| | Lite-Mono [51] | 2023 | 1 | M | 0.107 | 0.765 | 4.561 | 0.183 | 0.886 | 0.963 | 0.983 | 5.0 |
| | PuriLight [4] | 2026 | 1 | M | 0.106 | 0.747 | 4.536 | 0.182 | **0.890** | **0.964** | **0.984** | 7.1 |
| | **Flex-Nano (Ours)** | 2026 | 1 | M | 0.110 | 0.794 | 4.678 | 0.184 | 0.878 | 0.961 | 0.983 | **0.7** |
| | **Flex-Small (Ours)** | 2026 | 1 | M | **0.104** | **0.713** | **4.458** | **0.179** | **0.890** | **0.964** | 0.983 | 2.8 |
| | Lite-Mono-8M [51] | 2023 | 1 | M | 0.101 | 0.729 | 4.454 | 0.178 | 0.897 | 0.965 | 0.983 | 11.2 |
| | GeoDepth [44] | 2025 | 1 | M | 0.100 | 0.694 | 4.381 | 0.176 | 0.897 | 0.966 | 0.984 | 11.9 |
| | TinyDepth [5] | 2024 | 1 | M | 0.096 | 0.665 | 4.249 | **0.171** | 0.904 | **0.968** | **0.985** | 13.3 |
| | **Flex-Medium (Ours)** | 2026 | 1 | M | 0.096 | **0.639** | 4.253 | 0.172 | 0.903 | **0.968** | **0.985** | **10.0** |
| | **Flex-Large (Ours)** | 2026 | 1 | M | **0.095** | 0.642 | **4.199** | **0.171** | **0.906** | **0.968** | 0.984 | 11.5 |
| | MonoVit [55] | 2022 | 1 | M | 0.099 | 0.708 | 4.372 | 0.175 | 0.900 | 0.967 | 0.984 | 59.7 |
| | FGTO [29] | 2024 | 1 | M+Se | 0.096 | 0.696 | 4.327 | 0.174 | 0.904 | 0.968 | **0.985** | 59.7 |
| | DSI-MonoViT [52] | 2025 | 1 | M | 0.096 | 0.711 | 4.321 | 0.172 | 0.905 | 0.968 | 0.984 | 59.7 |
| | Self-Distillation [49] | 2025 | 1 | M | 0.095 | 0.619 | 4.156 | 0.169 | 0.905 | **0.969** | **0.985** | 64.5 |
| | **Flex-X-Large (Ours)** | 2026 | 1 | M | **0.093** | **0.605** | **4.114** | **0.167** | **0.910** | **0.969** | **0.985** | **24.6** |
| Cityscapes | ManyDepth [42] | 2021 | 2(-1,0) | M | 0.114 | 1.193 | 6.223 | 0.170 | 0.875 | 0.967 | 0.989 | 12.1 |
| | DynamicDepth [10] | 2022 | 2(-1,0) | M+Se | 0.103 | 1.000 | 5.867 | 0.157 | 0.895 | 0.974 | 0.991 | 18.5+ |
| | PuriLight [4] | 2026 | 1 | M | 0.099 | 1.014 | 5.892 | 0.153 | 0.901 | 0.976 | 0.992 | 5.7 |
| | **Flex-Nano (Ours)** | 2026 | 1 | M | 0.107 | 1.261 | 6.133 | 0.164 | 0.893 | 0.971 | 0.989 | **0.6** |
| | **Flex-Small (Ours)** | 2026 | 1 | M | 0.100 | 1.078 | 5.813 | 0.153 | 0.904 | 0.975 | 0.991 | 2.2 |
| | **Flex-Medium (Ours)** | 2026 | 1 | M | 0.089 | **0.885** | 5.358 | 0.143 | 0.917 | 0.979 | **0.993** | 8.0 |
| | **Flex-Large (Ours)** | 2026 | 1 | M | **0.087** | 0.911 | **5.310** | **0.139** | **0.924** | **0.981** | **0.993** | 9.2 |
| | ProDepth [43] | 2024 | 2(-1,0) | M | 0.095 | 0.876 | 5.531 | 0.146 | 0.908 | 0.978 | **0.993** | 35.5 |
| | MonoViT [55] | 2022 | 1 | M | 0.114 | 1.238 | 6.589 | 0.174 | 0.860 | 0.965 | 0.990 | 47.7 |
| | FGTO [29] | 2024 | 1 | M+Se | 0.096 | 0.930 | 5.806 | 0.152 | 0.905 | 0.976 | 0.992 | 47.7 |
| | Self-Distillation [49] | 2025 | 1 | M | 0.118 | 1.091 | 6.076 | 0.173 | 0.854 | 0.960 | 0.988 | 47.7+ |
| | DSI-MonoViT [52] | 2025 | 1 | M | 0.088 | **0.872** | 5.410 | 0.141 | 0.918 | 0.981 | **0.993** | 47.7 |
| | **Flex-X-Large (Ours)** | 2026 | 1 | M | **0.086** | 0.877 | **5.268** | **0.137** | **0.926** | **0.982** | **0.993** | **19.6** |

## 4 Experiments

### 4.1 Datasets

Following standard evaluation protocols, we validate our proposed method across three benchmark datasets. Specifically, the KITTI [12] and Cityscapes [6] datasets are utilized as domain-specific driving scenarios to evaluate the performance of our model within dynamic urban environments. To assess the generalization capabilities of FlexDepth, we conduct zero-shot evaluation on the Make3D dataset [35]. It is important to note that all training is performed using unlabeled video frames exclusively. Our approach does not rely on any auxiliary data, such as semantic labels, optical flow, or textual descriptions, ensuring an entirely self-supervised learning pipeline.

**KITTI [12] Dataset**. We adhere to the Eigen split [8] following established practices [14, 56], with 39,180 training images, 4,424 validation images, and 697 test images at a resolution of 640×192.

**Cityscapes [6] Dataset**. This dataset comprises a variety of dynamic scenarios and functions as the principal benchmark for evaluating dynamic models. We use 58,335 training images and 1,525 testing images provided by [10] at a resolution of 416×128.

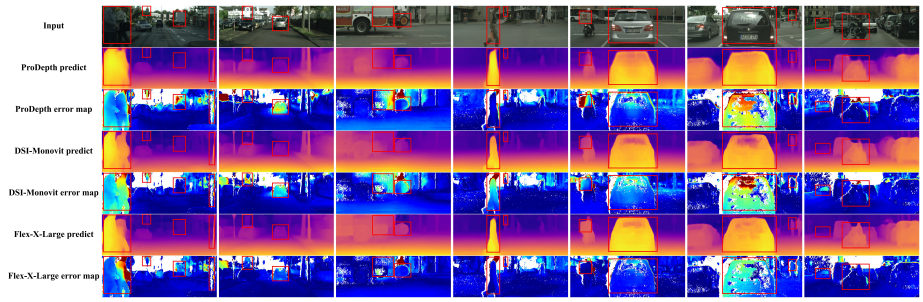


**Fig. 3: Qualitative evaluation on Cityscapes [6].** Our method is compared to dynamic scene handlers DSI [52] and ProDepth [43].In the error maps, blue indicates smaller errors and brighter red indicates larger ones. Our FlexDepth framework consistently yields more accurate depth with significantly reduced errors.

**Make3D [35] Dataset**. It comprises 134 test images of outdoor scenes, commonly used to evaluate MDE generalization. To assess our method's generalization capability, we directly apply the KITTI-trained model (640×192) to this dataset and report detailed evaluation metrics.

### 4.2 Specific Details in Training

We train all models using the NAdam optimizer [7]. Both datasets undergo a two-stage training process. In the first stage, the depth prediction network and PoseNet are trained. The second stage focuses on training the dynamic region processing capability. For the KITTI [12] dataset, the two stages are trained for 30 and 20 epochs respectively, while for the Cityscapes dataset, they are trained for 30 and 10 epochs respectively. We employ distinct learning rate schedulers for each phase: Stage 1 utilizes a step scheduler with a step size of 15 and a decay factor of 0.1, whereas Stage 2 adopts an ExponentialLR scheduler with a decay factor of 0.8. For the Nano/Small models with a batch size of 12, the initial learning rates for the encoder and decoder are set to 5e-5 and 1e-4, respectively. For the Medium/Large/X-Large models with a batch size of 6, both initial learning rates are set to 5e-5. All experiments are conducted using NVIDIA GPUs (RTX 4090 and RTX 2080 Ti). On the RTX 4090, our largest model (X-Large) requires approximately 10 hours to complete 50 epochs on the KITTI [12] dataset.

### 4.3 Experiment Results

Table 1 presents the quantitative results on the KITTI [12] and Cityscapes [6] datasets. The evaluation metrics adopted are the seven commonly used indicators proposed in [9], namely Abs Rel, Sq Rel, RMSE, RMSE log, $\delta < 1.25$, $\delta < 1.25^2$, and $\delta < 1.25^3$. We compare representative multi-scale models ranging from lightweight to heavy configurations. For the KITTI [12] dataset, models

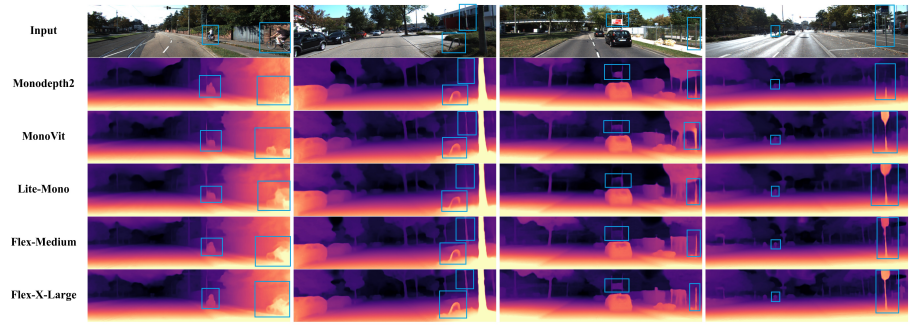


**Fig. 4: Qualitative comparison on KITTI [12].** Our method is compared with Monodepth2 [14], MonoVit [55], and Lite-Mono [51]. The proposed FlexDepth demonstrates superior performance in preserving fine details and structural integrity, particularly in challenging scenarios with complex object boundaries and textureless regions.

are categorized into three scales (0–9.9, 10.0–20.0, and above 20.0 GFLOPs). Since methods on the Cityscapes dataset are generally more complex, we define two scales (0–19.0 and above 19.0 GFLOPs). As shown in the Table 1, our approach achieves the lowest computational cost and state-of-the-art performance across all scales. Our smallest model, Flex-Nano, with only 0.7G FLOPs, delivers superior results compared to Lite-Mono [51]'s Tiny (2.8G) and Small (4.7G) versions. In the large-model category, our Flex-X-Large sets new state-of-the-art performance on both datasets.

The qualitative results on the two datasets are presented in Fig. 3 and Fig. 4, respectively. It can be observed that Flex-Medium and Flex-X-Large exhibit extensive receptive fields along with finer object contours and detailed textures. Particularly in highly dynamic scenarios, our model demonstrates strong adaptability, highlighting its superiority. As summarized in Table 5, we conduct a decoupled evaluation of dynamic regions and static backgrounds. The results demonstrate that, compared to existing methods relying on heavy backbones or complex network architectures, our Flex-X-Large variant achieves a substantial performance boost in dynamic areas while maintaining the lowest computational overhead. Notably, our approach achieves these results without any reliance on auxiliary information, such as pixel-wise motion fields or pseudo-depth maps. This demonstrates the inherent superiority of our architecture in handling complex, non-rigid scenes through a purely self-supervised pipeline. We provide a comparison with the foundation model in the Appendix.

**Zero-Shot Cross-Dataset Generalization.** To evaluate the zero-shot generalization capability, we conducted a cross-dataset evaluation on the Make3D [35] benchmark using a model trained exclusively on the KITTI [12] dataset. As shown in Table 3, FlexDepth demonstrates superior cross-dataset generalization. This exceptional transfer learning performance underscores the robustness and domain-invariant characteristics of the representations learned by our model, validating its potential applicability across diverse real-world scenarios.

**Table 2:** Ablation experiments on model architectures. Legend: $\mathcal{C}$: ✓=HPB, ○=HEB. $\mathcal{U}$: ✓=dynamic upsampling, ○=bilinear. $\mathcal{P}$: ✓=inverted head, ○=conventional head. Dynamic: ✓=static-dynamic decoupled mask $M$, ○=fixed threshold, ×=w/o stage-2. F.=FLOPs, P.=Params..

| ID | Decoder $\mathcal{C}$ | Decoder $\mathcal{U}$ | Decoder $\mathcal{P}$ | Dynamic | F.(G) | P.(M) | Depth Error ↓ Abs Rel | Sq Rel | RMSE | RMSE log | Depth Accuracy ↑ $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ○ | ✓ | ✓ | ✓ | **24.0** | **32.0** | **0.093** | 0.614 | 4.188 | 0.169 | 0.907 | **0.969** | **0.985** |
| 2 | ✓ | ○ | ✓ | ✓ | 24.3 | 32.3 | 0.094 | 0.627 | 4.204 | 0.169 | 0.906 | 0.968 | **0.985** |
| 3 | ✓ | ✓ | ○ | ✓ | 24.3 | 32.3 | 0.094 | 0.631 | 4.205 | 0.170 | 0.906 | 0.968 | **0.985** |
| 4 | ✓ | ✓ | ✓ | × | 24.6 | 32.3 | 0.100 | 0.729 | 4.431 | 0.175 | 0.900 | 0.967 | 0.984 |
| 5 | ✓ | ✓ | ✓ | ○ | 24.6 | 32.3 | **0.093** | 0.629 | 4.195 | 0.170 | 0.909 | 0.968 | **0.985** |
| 6 | ✓ | ✓ | ✓ | ✓ | 24.6 | 32.3 | **0.093** | **0.605** | **4.114** | **0.167** | **0.910** | **0.969** | **0.985** |

### 4.4 Ablation Experiment

The ablation results for all components are summarized in Table 2. For consistency, all experiments are conducted using the Flex-X-Large variant under identical training protocols.

**Impact of Post-Convolution Modules** (ID 1 vs. 6): As observed, employing the HEB-based post-convolution module slightly reduces computational overhead but results in a marginal performance degradation. This suggests that larger models are better suited for the HPB structure.

**Dynamic Upsampling** (ID 2 vs. 6): Traditional approaches relying on bilinear upsampling result in a moderate performance degradation, despite a marginal reduction in computational cost. This highlights the advantages of our proposed dynamic upsampling method: by adaptively determining optimal sampling locations from the original feature maps based on semantic content, our approach achieves more precise depth predictions.

**Inverted Prediction Head Order** (ID 3 vs. 6): Reversing the sequence of the prediction heads proves more beneficial for large-scale models. While this adjustment marginally increases computational cost by 0.2 GFLOPs, it yields a significant performance gain. This reinforces our rationale that high-channel features maintain superior structural and semantic consistency after upsampling, thereby enabling subsequent prediction convolutions to generate high-quality outputs with greater accuracy.

**Two-Stage Training Efficiency** (ID 4, 5, vs. 6): Excluding the second-stage processing for dynamic environments leads to a severe performance drop. This underscores the superiority of our Two-Stage Static-Dynamic Decoupled Training strategy. Compared to a fixed-threshold approach, our proposed static-dynamic decoupled mask $M$ provides higher precision and demonstrates better adaptability across a broader range of environments.

### 4.5 Complexity and Speed Evaluation

Table 4 presents computational cost and inference speed, where GPU benchmarks (RTX 4090, V100) use batch size 16 and mobile platforms (Snapdragon

**Table 3:** Quantitative Results on the Make3D dataset [35].

| Method | Abs Rel | Sq Rel | RMSE | RMSE log |
|---|---|---|---|---|
| Monodepth2 [14] | 0.322 | 3.589 | 7.417 | 0.163 |
| Lite-Mono-Tiny [51] | 0.318 | 3.217 | 7.234 | 0.164 |
| Lite-Mono-Small [51] | 0.310 | 2.878 | 6.842 | 0.160 |
| Lite-Mono [51] | 0.305 | 3.060 | 6.981 | 0.158 |
| Lite-Mono-8m [51] | 0.309 | 3.145 | 7.016 | 0.158 |
| GeoDepth [44] | 0.296 | **2.750** | 6.735 | 0.153 |
| **Flex-Small (Ours)** | 0.315 | 3.127 | 6.865 | 0.159 |
| **Flex-Medium (Ours)** | 0.292 | 2.864 | 6.686 | 0.152 |
| **Flex-X-Large(Ours)** | **0.279** | 2.799 | **6.599** | **0.144** |

**Table 4:** Speed at 640×192 resolution (SD 8Elite: Snapdragon 8 Elite ).

| Method | Full Model ↓ | | Speed (FPS) ↑ | | |
|---|---|---|---|---|---|
| | Params(M) | FLOPs(G) | 4090 | V100 | SD 8Elite |
| Lite-Mono-Tiny [51] | 2.143 | 2.842 | 1048 | 312 | 15.9 |
| Lite-Mono-Small [51] | 2.472 | 4.746 | 816 | 298 | 10.7 |
| Lite-Mono [51] | 3.069 | 5.032 | 781 | 295 | 10.3 |
| Lite-Mono-8M [51] | 8.766 | 11.212 | 592 | 223 | 5.6 |
| MonoVit [55] | 33.631 | 59.679 | 265 | 126 | 1.9 |
| **Flex-Nano(Ours)** | **1.522** | **0.718** | **1583** | **774** | **37.6** |
| **Flex-Small(Ours)** | 6.059 | 2.776 | 1215 | 606 | 18.6 |
| **Flex-Medium(Ours)** | 12.722 | 9.994 | 510 | 231 | 5.8 |
| **Flex-Large(Ours)** | 15.203 | 11.519 | 467 | 216 | 5.2 |
| **Flex-X-Large(Ours)** | 32.269 | 24.563 | 321 | 130 | 3.0 |

**Table 5:** Dynamic-scene depth estimation comparison on Cityscapes [6]. Our method uses no auxiliary cues (pixel motion, pseudo depth) and is most efficient.

| Method | Extra Info | | | Dynamic region | | Static region | | FLOPs(G)↓ |
|---|---|---|---|---|---|---|---|---|
| | Pix.Mot. | Pseu.Dep. | Frames | Abs Rel↓ | $\delta < 1.25$ ↑ | Abs Rel↓ | $\delta < 1.25$ ↑ | |
| ProDepth [43] | - | - | 2 | 0.115 | 0.884 | 0.092 | 0.911 | 35.5 |
| Mining-BrNet [30] | ✓ | ✓ | 1 | 0.119 | 0.872 | **0.080** | 0.922 | 27.0 |
| DSI-MonoVit [52] | - | - | 1 | 0.097 | 0.918 | 0.086 | 0.919 | 47.7 |
| **Flex-X-Large (Ours)** | - | - | 1 | **0.091** | **0.935** | 0.085 | **0.925** | **19.6** |

8 Elite) use batch size 1 in performance mode. Our small model achieves improvements while maintaining strong efficiency. Medium and large models involve slightly higher costs but significantly enhance performance. We provide a lightweight model for resource-constrained scenarios and a more accurate alternative for abundant computational resources, maintaining real-time inference without substantially increasing demands.

### 4.6 Comparison with Foundation Depth Models

Recent monocular depth foundation models [45, 46] have demonstrated impressive zero-shot generalization, raising the question of whether self-supervised methods trained on in-domain data still provide practical advantages. We address this by comparing FlexDepth with relative depth model Depth Anything V2 (DA2) [46] under a unified evaluation protocol.

**Fair Comparison Protocol.** To ensure a fair comparison, we unify the evaluation protocols of foundation models and self-supervised methods on KITTI [12].

**Depth Alignment.** Self-supervised methods like [14, 51] typically use per-image median scaling to align the predicted disparity to metric depth: $\hat{D}_i = D_i \cdot \frac{\text{median}(D^{\text{gt}})}{\text{median}(D)}$, where $D_i = 1/\text{disp}_i$, $D^{\text{gt}}$ is the ground-truth depth, and $\hat{D}_i$ is the aligned depth. Foundation models like [45, 46] use least-squares. Specifically, for each image $\frac{1}{\hat{D}_i} = s \cdot \text{disp}_i + t$, $[s, t] = \arg\min_{s,t} \sum_j \left\| s \cdot \text{disp}_j + t - \frac{1}{D_j^{\text{gt}}} \right\|^2$. Here $s, t$ are the scale and shift over valid pixels $j$.

**Table 6: Comparison with DA2 [46] on KITTI [12] Eigen [8] split (LS alignment + improved GT).** DA2 results are reproduced using official weights. †: Re-evaluation of our model under the this protocol for fair comparison, without introducing any new training or experiments. Full results in Appendix.

| Method | Type | Params | GFLOPs | Resolution | Abs Rel↓ | $\delta < 1.25$↑ |
|---|---|---|---|---|---|---|
| DA2 (ViT-L) | Zero-Shot | 335M | 1947 | 1722×518 | 0.070 | **0.956** |
| DA2 (ViT-S) | Zero-Shot | 25M | 137 | 1722×518 | 0.077 | 0.944 |
| DA2 (ViT-L) | Zero-Shot | 335M | 276 | 644×196 | 0.092 | 0.915 |
| DA2 (ViT-S) | Zero-Shot | 25M | 19 | 644×196 | 0.110 | 0.881 |
| Flex-X-Large† | Self-supervised | 32M | 25 | 640×192 | **0.063** | 0.952 |

**Ground Truth.** Foundation models are typically evaluated on the improved KITTI [12] ground truth [38] with dense multi-frame stereo completion, whereas self-supervised methods usually use the sparse LiDAR-based. We therefore additionally evaluate FlexDepth under the foundation-model protocol (LS alignment + improved GT). Full details are provided in the Appendix.

**Discussion.** Table 6 reports results under the matched protocol (LS alignment + improved GT). DA2 [46] demonstrates strong zero-shot generalization and excels at fine details ( $\delta < 1.25$ 0.956 vs. 0.952), but is sensitive to distant objects (see Appendix). In contrast, FlexDepth, self-supervised trained on KITTI [12], achieves superior accuracy (Abs Rel 0.063 vs. 0.070) with far fewer parameters and FLOPs. This suggests that in-domain self-supervision still offers practical value particularly for resource-constrained deployment complementing the strengths of large foundation models.

## 5 Conclusion and Discussion

This paper presents FlexDepth, a scale-driven family of flexible self-supervised monocular depth estimation models specifically engineered for robust driving perception. FlexDepth introduces a Scale-Driven Decoder (SDD), which adaptively selects its constitutive components to enhance feature fusion and generate high-precision depth maps. Furthermore, this paper proposes a two-stage static-dynamic decoupling training strategy designed to isolate dynamic driving entities from the static background. By performing independent confidence assessments, our approach achieves superior robustness in highly dynamic environments. Extensive evaluations demonstrate that FlexDepth achieves state-of-the-art performance across multiple autonomous driving benchmarks. Its minimal computational footprint allows for seamless deployment on edge devices, while its exceptional zero-shot generalization.

**Limitations.** Since FlexDepth is designed for robust driving perception, its applicability to man-made structures indoors with irregular textures and close-range occlusions is limited.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant 62332016.

## References

1. Aleotti, F., Tosi, F., Poggi, M., Mattoccia, S.: Generative adversarial networks for unsupervised monocular depth prediction. In: Proceedings of the European conference on computer vision (ECCV) workshops. pp. 0–0 (2018)
2. Bangunharcana, A., Magd, A., Kim, K.S.: Dualrefine: Self-supervised depth and pose estimation through iterative epipolar sampling and refinement toward equilibrium. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 726–738 (2023)
3. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
4. Chen, Y., Zhang, L., Chu, X., Zhang, T.: Purilight: A lightweight shuffle and purification framework for monocular depth estimation. arXiv preprint arXiv:2602.11066 (2026)
5. Cheng, Z., Zhang, Y., Yu, Y., Song, Z., Tang, C.: Tinydepth: Lightweight self-supervised monocular depth estimation based on transformer. Engineering Applications of Artificial Intelligence **138**, 109313 (2024)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
7. Dozat, T.: Incorporating nesterov momentum into adam (2016)
8. 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)
9. 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)
10. Feng, Z., Yang, L., Jing, L., Wang, H., Tian, Y., Li, B.: Disentangling object motion and occlusion for unsupervised multi-frame monocular depth. In: European Conference on Computer Vision. pp. 228–244. Springer (2022)
11. Garg, R., Bg, V.K., Carneiro, G., Reid, I.: Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: European conference on computer vision. pp. 740–756. Springer (2016)
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361 (2012). https://doi.org/10.1109/CVPR.2012.6248074
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: ICCV. pp. 3828–3838 (2019)

15. GonzalezBello, J.L., Kim, M.: Forget about the lidar: Self-supervised depth estimators with med probability volumes. Advances in Neural Information Processing Systems **33**, 12626–12637 (2020)
16. 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)
17. Guo, H., Zhu, H., Peng, S., Lin, H., Yan, Y., Xie, T., Wang, W., Zhou, X., Bao, H.: Multi-view reconstruction via sfm-guided monocular depth estimation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5272–5282 (2025)
18. Hui, T.W.: Rm-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1675–1684 (2022)
19. Jocher, G., Qiu, J., Chaurasia, A.: Ultralytics YOLO (Jan 2023), https://github.com/ultralytics/ultralytics
20. 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)
21. Ke, B., Qu, K., Wang, T., Metzger, N., Huang, S., Li, B., Obukhov, A., Schindler, K.: Marigold: Affordable adaptation of diffusion-based image generators for image analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
22. 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)
23. Lee, S., Im, S., Lin, S., Kweon, I.S.: Learning monocular depth in dynamic scenes via instance-aware projection consistency. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 1863–1872 (2021)
24. Li, B., Huang, Y., Liu, Z., Zou, D., Yu, W.: Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12663–12673 (2021)
25. Li, H., Gordon, A., Zhao, H., Casser, V., Angelova, A.: Unsupervised monocular depth learning in dynamic scenes. In: Conference on Robot Learning. pp. 1908–1917. PMLR (2021)
26. Liu, W., Lu, H., Fu, H., Cao, Z.: Learning to upsample by learning to sample. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6027–6037 (2023)
27. Luginov, A., Shahzad, M.: Nimbled: Enhancing self-supervised monocular depth estimation with pseudo-labels and large-scale video pre-training. In: European Conference on Computer Vision. pp. 235–251. Springer (2024)
28. Lyu, X., Liu, L., Wang, M., Kong, X., Liu, L., Liu, Y., Chen, X., Yuan, Y.: Hr-depth: High resolution self-supervised monocular depth estimation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 2294–2301 (2021)
29. Moon, J., Bello, J.L.G., Kwon, B., Kim, M.: 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. pp. 10519–10529 (2024)
30. Nguyen, H.C., Wang, T., Alvarez, J.M., Liu, M.: Mining supervision for dynamic regions in self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10446–10455 (2024)

31. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
33. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12179–12188 (2021)
34. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
35. 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)
36. 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)
37. Sun, Y., Hariharan, B.: Dynamo-depth: Fixing unsupervised depth estimation for dynamical scenes. Advances in Neural Information Processing Systems **36**, 54987–55005 (2023)
38. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant cnns. In: 2017 international conference on 3D Vision (3DV). pp. 11–20. IEEE (2017)
39. Wang, C., Buenaposada, J.M., Zhu, R., Lucey, S.: Learning depth from monocular videos using direct methods. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2022–2030 (2018)
40. Wang, P., Liu, S., Li, Q., Yi, Y., Wang, J.: Mg-mono: A lightweight multi-granularity method for self-supervised monocular depth estimation. Pattern Recognition p. 112223 (2025)
41. Watson, J., Firman, M., Brostow, G.J., Turmukhambetov, D.: Self-supervised monocular depth hints. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2162–2171 (2019)
42. Watson, J., Mac Aodha, O., Prisacariu, V., Brostow, G., Firman, M.: The temporal opportunist: Self-supervised multi-frame monocular depth. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1164–1174 (2021)
43. Woo, S., Lee, W., Kim, W.J., Lee, D., Lee, S.: Prodepth: Boosting self-supervised multi-frame monocular depth with probabilistic fusion. In: European Conference on Computer Vision. pp. 201–217. Springer (2024)
44. Wu, H., Gu, S., Duan, L., Li, W.: Geodepth: From point-to-depth to plane-to-depth modeling for self-supervised monocular depth estimation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 11525–11535 (2025)
45. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10371–10381 (2024)
46. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. Advances in Neural Information Processing Systems **37**, 21875–21911 (2024)

47. Yang, Z., Wang, P., Xu, W., Zhao, L., Nevatia, R.: Unsupervised learning of geometry from videos with edge-aware depth-normal consistency. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
48. Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., Chen, X., Shen, C.: Metric3d: Towards zero-shot metric 3d prediction from a single image. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9043–9053 (2023)
49. Yu, T., Pan, S., Chen, W., Tian, Z., Wang, Z., Yu, F.R., Leung, V.C.: Exploiting the potential of self-supervised monocular depth estimation via patch-based self-distillation. IEEE Internet of Things Journal (2025)
50. Zhan, H., Garg, R., Weerasekera, C.S., Li, K., Agarwal, H., Reid, I.: Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 340–349 (2018)
51. Zhang, N., Nex, F., Vosselman, G., Kerle, N.: 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. pp. 18537–18546 (2023)
52. Zhang, T., Zhang, L., Chu, X., Zhu, Z., Liu, Y.A.: Depth self-inhibition: An advanced training framework for self-supervised monocular depth estimation in dynamic scenes. Authorea Preprints (2025)
53. Zhang, W., Liu, H., Li, B., He, J., Qi, Z., Wang, Y., Zhao, S., Yu, X., Zeng, W., Jin, X.: Hybrid-grained feature aggregation with coarse-to-fine language guidance for self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6678–6692 (2025)
54. Zhang, Z., Cui, Z., Xu, C., Jie, Z., Li, X., Yang, J.: Joint task-recursive learning for semantic segmentation and depth estimation. In: Proceedings of the European conference on computer vision (ECCV). pp. 235–251 (2018)
55. Zhao, C., Zhang, Y., Poggi, M., Tosi, F., Guo, X., Zhu, Z., Huang, G., Tang, Y., Mattoccia, S.: Monovit: Self-supervised monocular depth estimation with a vision transformer. In: 2022 international conference on 3D vision (3DV). pp. 668–678. IEEE (2022)
56. 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)
57. Zhou, Z., Fan, X., Shi, P., Xin, Y., Duan, D., Yang, L.: Recurrent multiscale feature modulation for geometry consistent depth learning. IEEE transactions on pattern analysis and machine intelligence **46**(12), 9551–9566 (2024)