



# ANYDEPTH: DEPTH ESTIMATION MADE EASY

Zeyu Ren<sup>1\*</sup> Zeyu Zhang<sup>2\*†</sup> Wukai Li<sup>2</sup> Qingxiang Liu<sup>3</sup> Hao Tang<sup>2‡</sup>

<sup>1</sup>The University of Melbourne <sup>2</sup>Peking University <sup>3</sup>Shanghai University of Engineering Science

\*Equal contribution. †Project lead. ‡Corresponding author: bjdxtanghao@gmail.com.

*“Simplicity is prerequisite for reliability.” — Edsger W. Dijkstra*



Figure 1: We present **AnyDepth**, a simple and efficient training framework for zero-shot monocular depth estimation, which achieves impressive performance across a variety of indoor and outdoor scenes.

## ABSTRACT

Monocular depth estimation aims to recover the depth information of 3D scenes from 2D images. Recent work has made significant progress, but its reliance on large-scale datasets and complex decoders has limited its efficiency and generalization ability. In this paper, we propose a lightweight and data-centric framework for zero-shot monocular depth estimation. We first adopt DINOv3 as the visual encoder to obtain high-quality dense features. Secondly, to address the inherent drawbacks of the complex structure of the DPT, we design the Simple Depth Transformer (SDT), a compact transformer-based decoder. Compared to the DPT, it uses a single-path feature fusion and upsampling process to reduce the computational overhead of cross-scale feature fusion, achieving higher accuracy while reducing the number of parameters by approximately 85%–89%. Furthermore, we propose a quality-based filtering strategy to filter out harmful samples, thereby reducing dataset size while improving overall training quality. Extensive experiments on five benchmarks demonstrate that our framework surpasses the DPT in accuracy. This work highlights the importance of balancing model design and data quality for achieving efficient and generalizable zero-shot depth estimation. Code: <https://github.com/AIGeeksGroup/AnyDepth>. Website: <https://aigeeksgroup.github.io/AnyDepth>.

## 1 INTRODUCTION

Monocular depth estimation is gaining increasing attention due to its wide range of downstream applications. Depth maps are not only used to measure scene distances (Bhat et al., 2023; 2021; Godard et al., 2017), but can also be embedded as conditional information within models in the 3D reconstruction (Wang et al., 2025b;c;a), generation (Zhang et al., 2023; Rombach et al., 2022; Poole et al., 2022; Mildenhall et al., 2021; Li et al., 2024a; Yang et al., 2023), and embodied AI (Wu et al., 2025; Huang et al., 2025a; Liu et al., 2025b;a; Huang et al., 2025b; Song et al., 2025; Ye et al., 2025; Huang et al., 2025c;d), providing complementary information to improve granularity and geometric

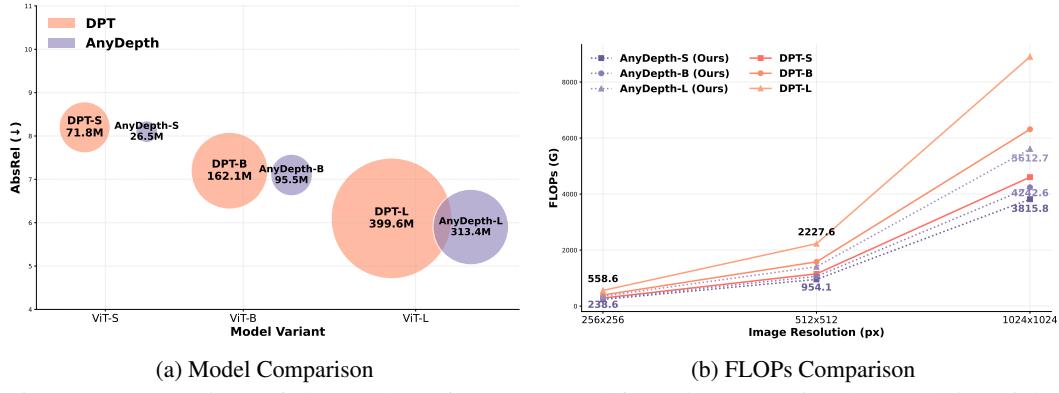


Figure 2: Comparison of the number of parameters (left) and computational complexity (right) of **AnyDepth** and DPT for different model sizes and input resolutions. Our method significantly reduces the number of model parameters and computational cost while maintaining competitive accuracy.

consistency. The MiDaS series (Ranftl et al., 2020; Birkl et al., 2023), through extensive and systematic experiments, compared the transfer performance of various pretrained vision transformers (such as ViT (Dosovitskiy et al., 2020), Swin (Liu et al., 2021), DINO (Oquab et al., 2023), and BeiT (Bao et al., 2021)) on monocular depth estimation tasks. DPT (Ranftl et al., 2021) has demonstrated impressive performance in various dense prediction tasks and is currently used as the decoder in mainstream models. DPT aims to achieve finer-grained predictions by fusing features at different scales. The Depth Anything series (Yang et al., 2024a;b) represents a typical data-driven approach, aiming to improve understanding and generalization capabilities of model for complex scenarios by leveraging massive datasets. These methods have significantly improved performance in zero-shot scenarios, demonstrating the potential of data scalability in the field of depth estimation.

However, We rethink the monocular depth estimation pipeline from both architectural and data-centric perspectives. From the architectural perspective, we observe that each Transformer layer in DPT requires a dedicated Re-assemble module to map features to different scales, followed by multiple alignment operations. This design introduces unnecessary complexity, large parameter counts, and slow inference speed. DPT uses fixed bilinear interpolation for upsampling, which lacks adaptability to local geometric structures and often leads to blurred edges and loss of fine spatial details. From the data perspective, purely data-driven approaches such as the Depth Anything series rely heavily on massive datasets. However, large-scale data collection is costly and inevitably introduces noisy samples that degrade training quality. Simply scaling model size and data quantity therefore provides limited gains and poor reproducibility.

Based on these findings and limitations, we aim to design a lightweight and efficient training framework that maintains competitive performance while being widely adopted by the research community (Fig. 2).

Specifically, our contributions are reflected in three aspects:

- We design a novel decoder that aligns and fuses features before restoring resolution through a one-shot reconstruction and upsampling. This architecture avoids multi-branch cross-scale alignment and repeated reconstruction, better preserving high-frequency details and geometric consistency.

- We analyze sample quality issues in deep learning datasets and proposed two metrics to quickly measure sample quality, which we then used to filter out low-quality samples. This reduced dataset size while improving overall data quality, demonstrating that our framework can achieve better performance with fewer resources.
- On multiple benchmarks, our framework achieves comparable accuracy and generalization to DPT with significantly fewer parameters and lower training overhead, demonstrating a superior efficiency-accuracy trade-off and academic reproducibility.

## 2 RELATED WORK

**Zero-Shot Monocular Depth Estimation.** To enable widespread use of depth images in real-world scenarios without relying on specific environments, zero-shot depth estimation has become a key research direction in recent years (Chen et al., 2016; Piccinelli et al., 2024; Chen et al., 2020; Yin et al., 2021). Due to the lack of strict geometric constraints on MDE, many zero-shot models learn to predict affine-invariant depth, i.e., recovering relative structure while maintaining scale and translation invariance (Ranftl et al., 2020; Yang et al., 2024a;b). For example, DiverseDepth (Yin et al., 2020) uses web images as training data to improve zero-shot generalization performance. MiDaS (Ranftl et al., 2020) proposed scale-shift-invariant losses to solve the ambiguity problem of different deep numerical representation methods of different datasets, so that the model can be trained on a large scale. In order to eliminate the inherent problems of the CNN backbone, the performance of Zero-Shot Monocular Depth Estimation was further improved by using the vision transformer architecture, such as DPT (Ranftl et al., 2021), Omnidata (Eftekhar et al., 2021), Depthformer (Li et al., 2023) and Zoepdeth (Bhat et al., 2023). Marigold (Ke et al., 2024) directly utilizes the standard diffusion model paradigm and stable diffusion pre-trained weights for fine-tuning to produce high-quality results. Depth Anything series (Yang et al., 2024a;b) used 62 million unlabeled images for larger-scale training. Geowizard (Fu et al., 2024) uses the high consistency between dense prediction tasks to jointly predict depth and normals. Lotus (He et al., 2024) analyzes the diffusion process to achieve single-step diffusion and speed up the inference process. Genpercept (Xu et al., 2024) uses experiments to prove that the diffusion model requires specific details to be optimized in dense prediction tasks.

**Decoder for Dense Prediction.** Currently, many methods for dense prediction tasks employ multi-scale feature fusion strategies to compensate for the lack of information from single-layer features (Lin et al., 2017; Liu et al., 2018; Tan et al., 2020; Chen et al., 2018; Ghiasi et al., 2019; Xu et al., 2021; Eigen & Fergus, 2015). FPN (Lin et al., 2017) proposes a top-down architecture where high-level semantic representations are successively merged with low-level features to enhance multi-scale features. (Lee et al., 2019) designed a multi-scale local plane guidance layer to more effectively guide the fusion of features at each layer to achieve performance improvement. Swin-Depth (Cheng et al., 2021) designs a lightweight multi-scale attention mechanism module to enhance the ability to learn global information at multiple scales. PVT (Wang et al., 2021) and Uformer (Wang et al., 2022) use a multi-scale pyramid decoder structure to capture long-range visual dependencies. DPT (Ranftl et al., 2021) utilizes the ViT (Dosovitskiy et al., 2020) backbone network to generate high-resolution features, thereby achieving finer-grained representation and improving prediction accuracy. However, multi-branch reassembly incurs significant overhead, especially in the case of high-resolution input.

## 3 THE PROPOSED METHOD

### 3.1 OVERVIEW

The proposed AnyDepth uses a pre-trained DINoV3 (Siméoni et al., 2025) encoder and SDT decoder; as shown in Fig.4, given an input image  $I$ , we extract multi-scale representations from four intermediate Transformer layers  $T^1, T^2, T^3, T^4$  and input them into the SDT head for depth reconstruction, thereby capturing different levels of detail and semantic information. These tokens are linearly projected onto a common dimension and fused to capture complementary semantic levels. The fused representations are then reshaped into feature maps and refined by a Spatial Detail Enhancer (SDE). Finally, a dense depth map is generated through two learnable Upsampler and head

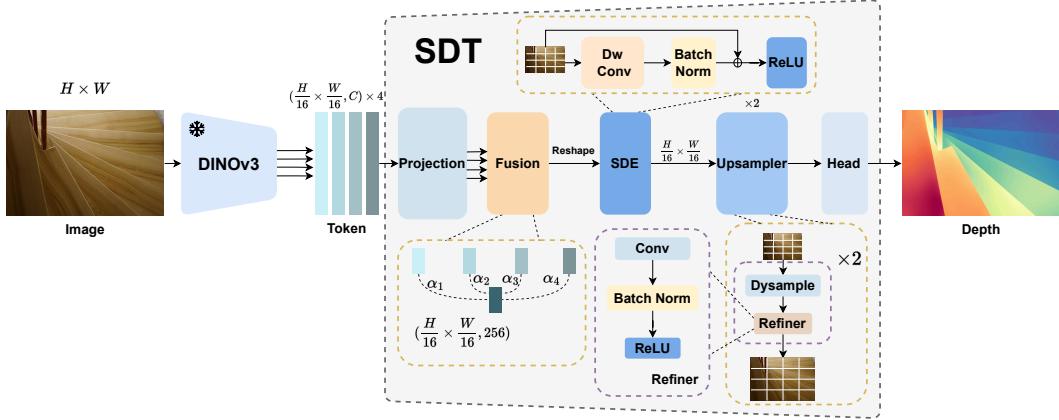


Figure 4: **AnyDepth architecture overview.** The input image is encoded into tokens by a frozen DINOv3 backbone network, then decoded by our lightweight SDT decoder. Tokens undergo only a single projection and weighted fusion. The Spatial Detail Enhancer (SDE) module ensures finer-grained predictions. The feature map is upsampled by an efficient and learnable upsampler dysample, and the depth is finally output by the head.

prediction. Our method differs from the Depth Anything series (Yang et al., 2024a;b) and DPT (Ranftl et al., 2021) in that we fuse tokens using only a single linear projection, followed by upsampling in a single path, without multi-branch cross-scale alignment, significantly reducing the number of parameters and computational overhead.

### 3.2 SIMPLE DEPTH TRANSFORMER (SDT)

Our decoder adopts a simple single-path fusion and reconstruction strategy, aiming to take advantage of the high-resolution feature of DINOv3 and further unleash its performance at high resolution. We first project the tokens extracted from the encoder into a 256-dimensional space using a linear layer followed by a GELU non-linearity (Hendrycks & Gimpel, 2016), which preserves sufficient informative content while substantially reducing the computational overhead in the subsequent decoding stages. For the class token, we keep the same processing as DPT (Ranftl et al., 2021), concatenate it with the spatial token, and then fuse it through the learnable projection.

**Fusion.** To fuse tokens from multiple layers of representation, we then employ a learnable weighted fusion strategy (Eq. 1).

Specifically, we assign a learnable scalar weight to each layer of tokens and normalize them using a softmax function to form a uniform probability distribution, preventing initial instability in training. This strategy enables the model to adaptively balance low-level structural details with high-level semantic information.

$$T = \sum_{i \in \mathcal{L}} \alpha_i \text{Proj}_i(T_i), \quad T_i \in \mathbb{R}^{N_p \times D}, \quad (1)$$

Where  $T_i$  denotes the token in layer  $i$  after projection, and contains  $N_p$  tokens of dimension  $D$ .

**Spatial Detail Enhancer.** After the fusion block, we reshape the sequence token output into a spatial feature map. Because the reorganized feature map lacks local continuity and, after multi-level fusion, easily obscures shallow texture details, which are crucial for dense prediction tasks such as depth estimation, we designed the Spatial Detail Enhancer. The SDE can be expressed by Eq. 2,

$$F' = \text{ReLU}(F + BN(DWConv_{3 \times 3}(F))), \quad F \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 256}. \quad (2)$$

We implement this operation first using a  $3 \times 3$  Depthwise convolution for local spatial modeling, followed by batch normalization. We then add the normalized response to the input feature  $F$  via a residual connection, and finally pass it through an activation layer.

**Upsampler.** In the upsampling stage, we abandon the commonly used bilinear interpolation, which easily blurs high-frequency details, and instead adopt a learnable dynamic sampler (Eq. 6).

Specifically, we use DySample (Liu et al., 2023) as the upsampler, which adaptively constructs an offset sampling grid based on the learned low-resolution features to adjust the sampling position, and then uses differentiable grid sampling to resample to high-resolution features. We first define three operators: the DySample block  $\mathcal{B}(\cdot)$ , the DySample stage  $\mathcal{S}(\cdot)$ , and the refinement block  $\mathcal{R}(\cdot)$ :

$$\mathcal{B}(X) = \text{ReLU}\left(\text{BN}\left(\text{Conv}_{3 \times 3}(\text{DySample}_{\times 2}(X))\right)\right), \quad (3)$$

$$\mathcal{S}(X) = \mathcal{B}(\mathcal{B}(X)), \quad (4)$$

$$\mathcal{R}(X) = \text{ReLU}\left(\text{BN}\left(\text{Conv}_{3 \times 3}(X)\right)\right). \quad (5)$$

Based on these definitions (Eq. 3, 4, 5), the complete upsampling process can be expressed as:

$$\mathcal{U}(X) = \mathcal{R}\left(\mathcal{S}\left(\mathcal{R}(\mathcal{S}(X))\right)\right), \quad (6)$$

In this way, the compact feature map of size  $H/16 \times W/16$  can be progressively upsampled back to the original resolution  $H \times W$ . We want to emphasize that we do not jump to  $H \times W$  all at once, but rather decompose the upsampling into two  $\times 4$  upsamplers, using four dysamples of scale 2. Single-stage  $\times 16$  upsampling forces the sampler to infer large offsets from very low-resolution features, which amplifies errors and destabilizes gradients. Our progressive design keeps the offsets small, inserting local refinement after each resampling, resulting in a model with better detail recovery capabilities.

### 3.3 SDT vs. DPT

A key difference between SDT and DPT (Ranftl et al., 2021) is the order of feature reassembly. DPT employs a reassemble-fusion strategy. Specifically, DPT first applies the reassemble module to the tokens extracted by each Transformer layer, mapping the tokens to feature maps of different scales. These feature maps are then fused in a cascade across scales, which inevitably introduces multiple branches and repeated cross-scale alignment overhead. In contrast, SDT employs a fusion-reassemble strategy, directly projecting and fusing groups of tokens. Only after this stage do we perform spatial reassembly and upsampling along a single path. This fusion-reassemble strategy avoids the high cost of per-layer token reassembly and feature map cross-scale alignment, making it more efficient and stable, especially when processing high-resolution inputs.

## 4 EXPERIMENTS

### 4.1 DATASETS AND METRICS

**Training Datasets.** We use five synthetic datasets covering various indoor and outdoor scenes for training. (1) *Hypersim* (Roberts et al., 2021) after filtering incomplete samples, we have approximately 39K. (2) *Virtual KITTI* (Cabon et al., 2020) we selected four scenes, totaling approximately 20K. (3) *BlendedMVS* (Yao et al., 2020) (4) *IRS* (Wang et al., 2019) (5) *TartanAir* (Wang et al.) As shown in Table 1, we only use 369K datasets for training. The far plane is set to 100 m. To improve the robustness and generalization of the model, we used data augmentation of flipping and rotation.

**Evaluation Datasets and Metrics.** For Zero-shot monocular depth estimation, we evaluate SDT using five datasets containing various scenes: NYUv2 (Silberman et al., 2012), KITTI (Geiger et al., 2013), ETH3D (Schops et al., 2017), ScanNet (Dai et al., 2017), and DIODE (Vasiljevic et al., 2019). We use the absolute mean relative error(AbsRel), i.e.,  $\frac{1}{M} \sum_{i=1}^M \frac{|\hat{d}_i - d_i|}{d_i}$ , where  $M$  is the total number of valid pixels,  $d_i$  denotes the ground truth, and  $\hat{d}_i$  is the predicted depth. We report accuracy thresholds  $\delta_\tau$ , which denote the fraction of pixels where the prediction and ground truth differ by less than a multiplicative factor  $\tau = 1.25$ .

### 4.2 IMPLEMENTATION DETAILS

Our setup differs slightly from Depth Anything V2 (Yang et al., 2024b). To better utilize the high-resolution features of DINOv3 (Siméoni et al., 2025), we increase the input image resolution to

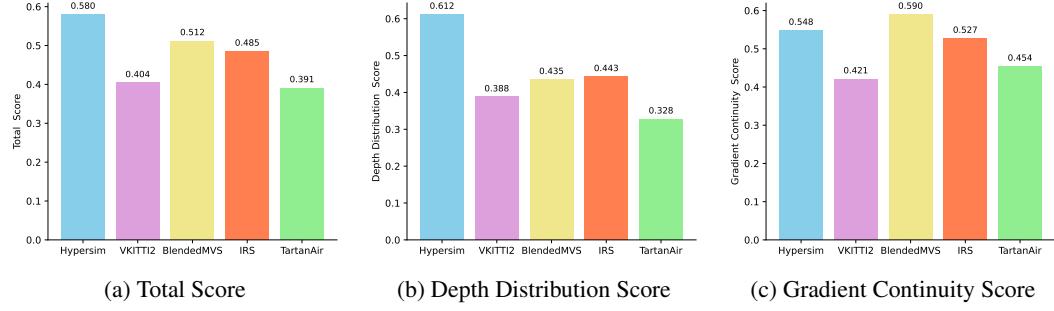


Figure 5: Dataset quality across the Total Score, Depth Distribution Score, and Gradient Continuity Score (higher is better).

$768 \times 768$ . The encoder is kept frozen throughout training, and we use features from four intermediate layers as decoder inputs: [2, 5, 8, 11] for DINOv3 S/16 and DINOv3 B/16, and [4, 11, 17, 23] for DINOv3 L/16. We perform simple regression to predict disparity  $d' = 1/d$ , where  $d'$  denotes disparity and  $d$  denotes depth. Both the input image and the groundtruth are normalized to  $[0, 1]$ . We follow the settings of Depth Anything v2 (Yang et al., 2024b) and use a scale- and shift-invariant loss  $\mathcal{L}_{ssi}$  and a gradient matching loss  $\mathcal{L}_{gm}$ , and the weight ratio of  $\mathcal{L}_{ssi}$  and  $\mathcal{L}_{gm}$  is set to 1 : 2. To stabilize optimization, we follow an optimization strategy similar to DINOv3 (Siméoni et al., 2025). We use AdamW with a base learning rate of  $1 \times 10^{-3}$ , a PolyLR scheduler with power 0.9, and a linear warm-up for the first two epochs. We train for a total of five epochs.

### 4.3 MAIN RESULTS

#### 4.3.1 RESULTS OF DATA CENTRIC LEARNING

We applied the metrics proposed in Section A.2 to all training datasets, with the results shown in Fig. 5. We observe that Hypersim performed well in both the Depth Distribution Score and Gradient Continuity Score, achieving the highest overall score. This indicates a relatively balanced depth distribution, smooth gradients, and a low concentration of noisy samples. In contrast, datasets containing outdoor samples, such as VKITTI2, BlendedMVS, and TartanAir, had significantly lower Depth Distribution Scores, indicating a more severe depth distribution. This is likely a common problem across all outdoor datasets. The low Gradient Continuity Score for VKITTI2 may be due to the presence of numerous fine-grained structures (*e.g.*, leaves) in the samples, resulting in abundant edges and severe gradient abruptness, which is considered noisy.

Following the methods described in Section A.2, we filtered the entire dataset. Specifically, we first filtered out samples whose valid depth values accounted for less than 20% of the total pixels. We then sorted the remaining samples based on the Depth Distribution Score and Gradient Continuity Score, filtering out the 20% with the lowest scores for each metric. The number of filtered samples for each dataset is shown in Table 1. For visualizations of low-quality samples, please see the A.3. The merged dataset contains 584K samples, of which approximately 369K are used for training and 215K are filtered out.

#### 4.3.2 QUANTITATIVE COMPARISONS

Table 2 reports quantitative comparison results for zero-shot affine-invariant depth estimation. Since the baselines in the Depth Anything series all use a DPT head, we primarily compare our proposed SDT decoder with the DPT under the same backbone settings.

Table 1: Dataset statistics of good and bad samples.

Dataset	Total	Good	Bad
Hypersim	39,648	26,912	12,736
VKITTI2	19,559	12,643	6,916
BlendedMVS	115,142	74,838	40,304
IRS	103,316	68,211	35,105
TartanAir	306,637	186,693	119,944
<b>Summary</b>	<b>584,302</b>	<b>369,297</b>	<b>215,005</b>

Table 2: Quantitative comparison of zero-shot affine-invariant depth estimation. Lower AbsRel values are better; higher  $\delta_1$  values are better. DINOv3 (Siméoni et al., 2025) uses the ViT-7B encoder, and Depth Anything v2 (DAv2) (Yang et al., 2024b) is trained on 62.6M datasets. For fair comparison, the baseline (DPT) uses a frozen DINOv3 encoder and DPT head, while our method replaces the DPT head with the proposed SDT. The bold numbers in the table refer to the best results between DPT and AnyDepth.

Method	Training Data↓	Encoder	#Params (M)↓	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
				AbsRel↓	$\delta_1 \uparrow$	AbsRel↓	$\delta_1 \uparrow$						
DINOv3	595K	ViT-7B	91.19	4.3	98.0	7.3	96.7	5.4	97.5	4.4	98.1	25.6	82.2
		ViT-S	71.8	5.3	97.3	7.8	93.6	14.2	85.1	—	—	7.3	94.2
		ViT-B	162.1	4.9	97.6	7.8	93.9	13.7	85.8	—	—	6.8	95.0
DAv2	62.6M	ViT-L	399.6	4.5	97.9	7.4	94.6	13.1	86.5	—	—	6.6	95.2
		ViT-S	71.8	8.4	93.3	10.8	89.1	12.7	92.0	8.3	93.5	26.0	71.4
		ViT-B	162.1	7.5	95.1	10.8	88.9	10.0	92.9	7.1	95.3	24.5	73.4
DPT	584K	ViT-L	399.6	6.1	<b>96.8</b>	8.9	92.5	13.0	94.9	6.0	97.0	23.4	<b>73.9</b>
		ViT-S	26.5	8.2	93.2	10.2	88.3	8.4	93.5	8.0	93.6	24.7	71.4
		ViT-B	95.5	7.2	95.0	9.7	90.1	<b>8.0</b>	94.5	6.8	95.6	23.6	72.7
<b>AnyDepth</b>	369K	ViT-L	313.4	<b>6.0</b>	<b>96.8</b>	<b>8.6</b>	<b>92.6</b>	9.6	<b>95.4</b>	<b>5.4</b>	<b>97.4</b>	<b>22.6</b>	73.6

Table 3: Comparison of zero-shot affine-invariant depth estimation with different encoders and decoders. Green cells indicate the best results within each method.

Method	Encoder	Decoder	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
			AbsRel↓	$\delta_1 \uparrow$	AbsRel↓	$\delta_1 \uparrow$	AbsRel↓	$\delta_1 \uparrow$	AbsRel↓	$\delta_1 \uparrow$	AbsRel↓	$\delta_1 \uparrow$
DAv2	ViT-B	DPT	5.8	96.2	<b>10.4</b>	89.1	8.8	94.6	6.2	95.3	<b>23.4</b>	73.8
		SDT	<b>5.6</b>	<b>96.4</b>	10.7	<b>89.6</b>	<b>7.5</b>	<b>95.8</b>	<b>6.1</b>	<b>95.4</b>	23.9	<b>73.9</b>
DAv3	ViT-L	DPT	<b>4.9</b>	96.9	<b>8.8</b>	<b>92.4</b>	6.9	95.9	5.0	<b>96.6</b>	22.5	74.6
		Dual-DPT	<b>4.9</b>	97.0	8.9	<b>92.4</b>	7.0	95.8	<b>4.9</b>	<b>96.6</b>	22.3	74.6
		SDT	<b>4.9</b>	<b>97.1</b>	8.9	<b>92.4</b>	<b>5.8</b>	<b>96.6</b>	5.0	<b>96.6</b>	<b>21.9</b>	<b>74.9</b>
VGGT	VGGT-1B	DPT	<b>4.8</b>	97.7	15.6	77.9	7.2	94.7	<b>4.6</b>	97.6	30.7	76.2
		SDT	<b>4.8</b>	<b>98.0</b>	<b>15.5</b>	<b>80.1</b>	<b>7.0</b>	<b>95.1</b>	<b>4.6</b>	<b>98.0</b>	<b>30.6</b>	<b>76.8</b>

While our approach does not yet surpass the state-of-the-art results reported by fully data-driven methods (*e.g.*, the Depth Anything series (Yang et al., 2024a;b) and DINOv3-7B (Siméoni et al., 2025), which require hundreds of millions of parameters or massive datasets), we emphasize that our entire AnyDepth is designed from a light-weight and simple perspective, focusing not only on model design but also on data quality and quantity. Inspired by the principles of data-centric learning, we conclude that our model can achieve superior performance even with a relatively small amount of high-quality data (369K).

SDT uses only 5–13M parameters and outperforms DPT with various encoder sizes. Our results show that SDT significantly reduces the number of parameters and training cost while maintaining comparable accuracy to DPT, and there is a slight improvement in inference speed (Fig. 3). AnyDepth provides a lightweight, efficient, and computationally friendly alternative.

#### 4.4 EFFICIENCY

We comprehensively evaluated efficiency advantages of AnyDepth. Compared to DPT, AnyDepth not only significantly reduces the number of parameters (Fig. 2a), but also shows that AnyDepth significantly reduces FLOPs by 37% when using models of varying sizes, particularly at high res-

Table 5: Decoder parameter comparison across different ViT backbones. Lower is better.

Decoder	ViT Backbone	Params (M)↓
DPT	ViT-S	50.83
	ViT-B	76.05
	ViT-L	99.58
SDT	ViT-S	<b>5.51</b>
	ViT-B	<b>9.45</b>
	ViT-L	<b>13.38</b>

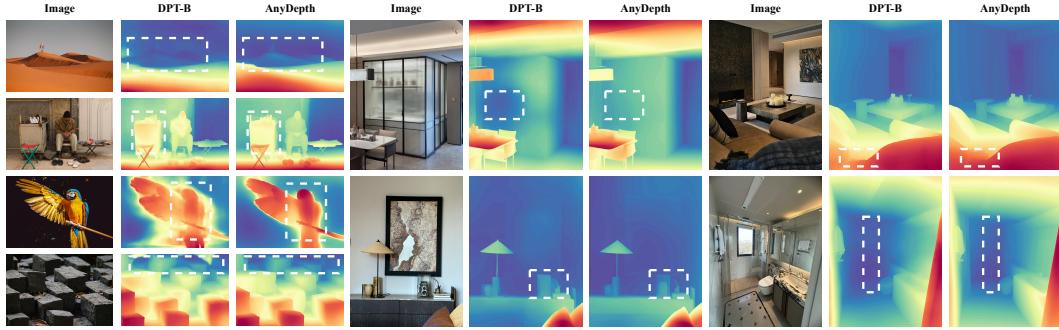


Figure 6: Qualitative results of zero-shot monocular depth estimation using **AnyDepth** of ViT-B and comparison with DPT-B.

Table 4: Multi-resolution efficiency comparison of SDT and DPT heads under a ViT-L encoder. Latency is averaged over 1000 runs on an NVIDIA H100 GPU. Lower is better.

Resolution	Decoder	FLOPs (G)↓	Latency (ms)↓
256×256	DPT	444.14	6.66 ± 0.22
	<b>SDT (Ours)</b>	<b>234.17</b>	<b>6.10 ± 0.33</b>
512×512	DPT	1776.56	24.65 ± 0.22
	<b>SDT (Ours)</b>	<b>936.70</b>	<b>23.17 ± 0.54</b>
1024×1024	DPT	7106.22	99.79 ± 0.79
	<b>SDT (Ours)</b>	<b>3746.79</b>	<b>93.09 ± 0.51</b>

olutions (Fig.2b). It also slightly improves inference speed (Fig.3). Furthermore, Average iteration time of AnyDepth during training is 10% shorter than that of DPT.

To explore the sources of these efficiency improvements, we further compared the efficiency of the proposed SDT decoder and DPT decoder under the same experimental settings. As shown in Tables 5 and Table 4, SDT consistently and significantly reduces the number of parameters and computational cost across different ViT backbone network sizes and input resolutions. Importantly, the reduction in model size did not affect runtime performance, as the inference latency of SDT is comparable to or even slightly faster than that of DPT.

#### 4.5 REAL WORLD EVALUATION

As shown in Fig. 7, We use the WHEELTEC R550 as the mobile platform for real-world evaluation. The robot is equipped with a Jetson Orin Nano 4GB as the on-board computing unit and an Astra Pro RGB-D camera as the perception unit. To evaluate its universality under various real-world conditions, we set up three different scenarios: a conference room, a corridor, and a rest area. Under the same encoder experimental setup, we used different decoders for real-world qualitative evaluation. As shown in Figure 10, the SDT decoder performs better than the DPT decoder, displaying clearer boundaries in complex areas.

Furthermore, we compared the efficiency performance of SDT and DPT on edge devices. As shown in Table 6, we compared the inference latency and throughput of the SDT and DPT decoders on the Jetson Orin Nano (4GB) at two input resolutions. At both 256×256 and 512×512 resolutions, SDT consistently outperforms DPT in terms of inference la-

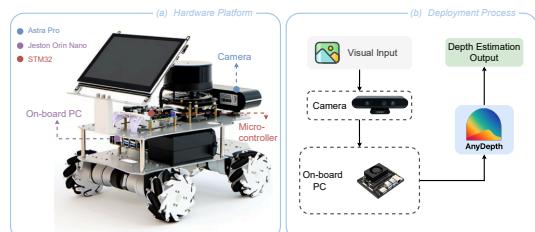


Figure 7: Hardware and Evaluation Pipeline for Real-World Experiments

Table 7: Peak GPU memory usage during inference at 256 × 256 resolution on Jetson Orin Nano (4GB).

Decoder	Peak Memory (MB)↓
DPT	589.5
<b>SDT (Ours)</b>	<b>395.2</b>

Table 6: Inference latency comparison of SDT and DPT decoders on a Jetson Orin Nano (4GB).

Resolution	Decoder	Latency (ms)↓	FPS↑
256×256	DPT	305.65	3.3
	<b>SDT (Ours)</b>	<b>213.35</b>	<b>4.7</b>
512×512	DPT	1107.64	0.9
	<b>SDT (Ours)</b>	<b>831.48</b>	<b>1.2</b>

Table 8: Ablation experiments of AnyDepth-B on five benchmarks. We report AbsRel (lower is better) and  $\delta_1$  (higher is better).

Method	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
	AbsRel↓	$\delta_1$ ↑	AbsRel↓	$\delta_1$ ↑						
w/o Filtering	9.5	91.1	15.4	77.3	14.0	91.2	8.3	93.5	25.0	71.1
Filtering	9.3	91.6	15.1	78.1	12.8	90.5	8.0	93.9	24.8	71.1
Filtering + SDE	8.8	92.4	14.7	79.6	11.5	91.0	7.9	94.1	24.3	71.1
<b>Filtering + SDE + Dysample</b>	<b>7.2</b>	<b>95.0</b>	<b>9.7</b>	<b>90.1</b>	<b>8.0</b>	<b>94.5</b>	<b>6.8</b>	<b>95.6</b>	<b>23.6</b>	<b>72.7</b>

tency and frame rate. As shown in Table 7, at 256×256 resolution, SDT requires approximately 33% less peak memory than the DPT decoder.

#### 4.6 ABLATION STUDY

We conducted ablation studies to validate our design. We used AnyDepth of ViT-B to progressively test our components, including data filtering, SDE, and DySample. As shown in the table 8, these ablation studies further support the effectiveness of data-centric learning in monocular depth estimation and demonstrate the detail enrichment capability of the SDE module and the additional gain of DySample compared to bilinear upsampling.

## 5 LIMITATIONS AND FUTURE WORK

While our work demonstrates advantages, it also has some limitations. First, the current pipeline has not been evaluated in large-scale fully supervised or fine-tuned settings. Second, further analysis of the dataset can be used to optimize the filtering strategy. In future work, we can extend our lightweight framework to a wider range of tasks, such as metric depth and normal estimation.

## 6 CONCLUSION

In this paper, we introduce AnyDepth, a simple and efficient-to-train framework for zero-shot monocular depth estimation. In our setup, a powerful self-supervised visual backbone paired with a single-path lightweight decoder is sufficient to achieve competitive performance without the need for large-scale, costly training. The goal of AnyDepth is not to surpass large-scale state-of-the-art methods, but rather to provide a more practical and academically valuable approach through its lightweight design and improved data quality.

## REFERENCES

- Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.
- Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4009–4018, 2021.
- Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

- Reiner Birlk, Diana Wofk, and Matthias Müller. Midas v3. 1—a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023.
- Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020.
- Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.
- Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29, 2016.
- Weifeng Chen, Shengyi Qian, David Fan, Noriyuki Kojima, Max Hamilton, and Jia Deng. Oasis: A large-scale dataset for single image 3d in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 679–688, 2020.
- Zeyu Cheng, Yi Zhang, and Chengkai Tang. Swin-depth: Using transformers and multi-scale fusion for monocular-based depth estimation. *IEEE Sensors Journal*, 21(23):26912–26920, 2021.
- Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10786–10796, 2021.
- David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pp. 2650–2658, 2015.
- Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *European Conference on Computer Vision*, pp. 241–258. Springer, 2024.
- Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.
- Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7036–7045, 2019.
- Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 270–279, 2017.
- Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bi-directional cascade network for perceptual edge detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3828–3837, 2019.
- Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124*, 2024.
- Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

- Ting Huang, Dongjian Li, Rui Yang, Zeyu Zhang, Zida Yang, and Hao Tang. Mobilevla-r1: Reinforcing vision-language-action for mobile robots. *arXiv preprint arXiv:2511.17889*, 2025a.
- Ting Huang, Zeyu Zhang, and Hao Tang. 3d-r1: Enhancing reasoning in 3d vlms for unified scene understanding. *arXiv preprint arXiv:2507.23478*, 2025b.
- Ting Huang, Zeyu Zhang, Yemin Wang, and Hao Tang. 3d coca: Contrastive learners are 3d captioners. *arXiv preprint arXiv:2504.09518*, 2025c.
- Ting Huang, Zeyu Zhang, Ruicheng Zhang, and Yang Zhao. Dc-scene: Data-centric learning for 3d scene understanding. *arXiv preprint arXiv:2505.15232*, 2025d.
- Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9492–9502, 2024.
- Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.
- Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20775–20785, 2024a.
- Pengzhi Li, Yikang Ding, Haohan Wang, Chengshuai Tang, and Zhiheng Li. The devil is in the edges: monocular depth estimation with edge-aware consistency fusion. *arXiv preprint arXiv:2404.00373*, 2024b.
- Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *Machine Intelligence Research*, 20(6):837–854, 2023.
- Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.
- Qingxiang Liu, Ting Huang, Zeyu Zhang, and Hao Tang. Nav-r1: Reasoning and navigation in embodied scenes. *arXiv preprint arXiv:2509.10884*, 2025a.
- Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768, 2018.
- Wenze Liu, Hao Lu, Hongtao Fu, and Zhiguo Cao. Learning to upsample by learning to sample. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6027–6037, 2023.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.
- Zeting Liu, Zida Yang, Zeyu Zhang, and Hao Tang. Evovla: Self-evolving vision-language-action model. *arXiv preprint arXiv:2511.16166*, 2025b.
- Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

- Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10106–10116, 2024.
- Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- Mengyang Pu, Yaping Huang, Yuming Liu, Qingji Guan, and Haibin Ling. Edter: Edge detection with transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1402–1412, 2022.
- René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.
- René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12179–12188, 2021.
- Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10912–10922, 2021.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 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.
- Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3260–3269, 2017.
- Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pp. 746–760. Springer, 2012.
- Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.
- Prerna Singh. Systematic review of data-centric approaches in artificial intelligence and machine learning. *Data Science and Management*, 6(3):144–157, 2023.
- Zirui Song, Guangxian Ouyang, Mingzhe Li, Yuheng Ji, Chenxi Wang, Zixiang Xu, Zeyu Zhang, Xiaoqing Zhang, Qian Jiang, Zhenhao Chen, et al. Maniplvm-r1: Reinforcement learning for reasoning in embodied manipulation with large vision-language models. *arXiv preprint arXiv:2505.16517*, 2025.
- Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5117–5127, 2021.
- Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790, 2020.
- Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019.
- Qiang Wang, Shizhen Zheng, Qingsong Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. Irs: A large naturalistic indoor robotics stereo dataset to train deep models for disparity and surface normal estimation. *arXiv preprint arXiv:1912.09678*, 2019.

- Weijie Wang, Donny Y Chen, Zeyu Zhang, Duochao Shi, Akide Liu, and Bohan Zhuang. Zpressor: Bottleneck-aware compression for scalable feed-forward 3dgs. *arXiv preprint arXiv:2505.23734*, 2025a.
- Weijie Wang, Yeqing Chen, Zeyu Zhang, Hengyu Liu, Haoxiao Wang, Zhiyuan Feng, Wenkang Qin, Zheng Zhu, Donny Y Chen, and Bohan Zhuang. Volsplat: Rethinking feed-forward 3d gaussian splatting with voxel-aligned prediction. *arXiv preprint arXiv:2509.19297*, 2025b.
- Weijie Wang, Jiagang Zhu, Zeyu Zhang, Xiaofeng Wang, Zheng Zhu, Guosheng Zhao, Chaojun Ni, Haoxiao Wang, Guan Huang, Xinze Chen, et al. Drivegen3d: Boosting feed-forward driving scene generation with efficient video diffusion. *arXiv preprint arXiv:2510.15264*, 2025c.
- Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 568–578, 2021.
- Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. in 2020 ieee. In *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916.
- Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17683–17693, 2022.
- Zhengri Wu, Yiran Wang, Yu Wen, Zeyu Zhang, Biao Wu, and Hao Tang. Stereoadapter: Adapting stereo depth estimation to underwater scenes. *arXiv preprint arXiv:2509.16415*, 2025.
- Guangkai Xu, Yongtao Ge, Mingyu Liu, Chengxiang Fan, Kangyang Xie, Zhiyue Zhao, Hao Chen, and Chunhua Shen. What matters when repurposing diffusion models for general dense perception tasks? *arXiv preprint arXiv:2403.06090*, 2024.
- Xianfa Xu, Zhe Chen, and Fuliang Yin. Monocular depth estimation with multi-scale feature fusion. *IEEE Signal Processing Letters*, 28:678–682, 2021.
- Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8254–8263, 2023.
- Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 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, 2024a.
- Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024b.
- Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry from videos with edge-aware depth-normal consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1790–1799, 2020.
- Angen Ye, Zeyu Zhang, Boyuan Wang, Xiaofeng Wang, Dapeng Zhang, and Zheng Zhu. Vla-r1: Enhancing reasoning in vision-language-action models. *arXiv preprint arXiv:2510.01623*, 2025.
- Wei Yin, Xinlong Wang, Chunhua Shen, Yifan Liu, Zhi Tian, Songcen Xu, Changming Sun, and Dou Renyin. Diversedepth: Affine-invariant depth prediction using diverse data. *arXiv preprint arXiv:2002.00569*, 2020.

Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 204–213, 2021.

Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. Data-centric artificial intelligence: A survey. *ACM Computing Surveys*, 57(5):1–42, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

## A APPENDIX

### A.1 LLM USE DECLARATION

Large Language Models (ChatGPT) were used exclusively to improve the clarity and fluency of English writing. They were not involved in research ideation, experimental design, data analysis, or interpretation. The authors take full responsibility for all content.

### A.2 DATA CENTRIC LEARNING

Although MiDaS (Ranftl et al., 2020) uses an affine-invariant loss to accommodate multi-dataset training, the varying degrees of noise and scale ambiguity introduced by these datasets can easily negatively impact training, especially in dense prediction tasks (Fig.8, 9). Inspired by data-centric learning (Singh, 2023; Zha et al., 2025), for the monocular depth estimation task and our setting, we believe that high-quality samples should possess two properties: (i) depth values should be evenly distributed throughout the image, rather than being overly concentrated within a specific range; and (ii) gradient magnitudes should vary slightly across continuous surfaces, while exhibiting more pronounced changes near object edges. Based on these two properties, we define two metrics to measure sample quality. These metrics aim to reduce low-quality samples, facilitate model training, and reduce dataset size and training cost.

#### A.2.1 DEPTH DISTRIBUTION SCORE

Some samples have depths that are primarily concentrated near or far, while other depth ranges are relatively small. As shown in Fig. 8, this phenomenon is common in outdoor datasets. This unbalanced depth distribution can cause the model to favor learning depth values within a specific range rather than the entire valid depth range, leading to unstable training and poor model generalization.

To quantify this phenomenon, we propose a *Depth Distribution Score* that evaluates how uniformly depth values are distributed across the available depth range. For a depth map  $D \in \mathbb{R}^{H \times W}$ , we divide the depth values into  $K$  bins of equal width, and we use  $K = 20$  by default to balance granularity and robustness.

**Chi-square Deviation** ( $S_{\chi^2}$ ). We measure the deviation from a uniform distribution using the chi-square statistic:

$$\chi^2 = \sum_{k=1}^K \frac{(n_k - \bar{n})^2}{\bar{n}}, \quad S_{\chi^2} = \exp\left(-\frac{\chi^2}{N}\right), \quad (7)$$

where  $n_k$  is the number of depth bins  $k$ ,  $\bar{n} = N/K$  is the expected number under a uniform distribution, and  $N$  is the total number of valid depth values. We use an exponential transformation to map the chi-squared statistic (Eq. 7) to  $[0, 1]$ , with higher scores indicating a more uniform distribution.

**Maximum Concentration Index** ( $S_{\text{conc}}$ ). To prevent excessive concentration in any single depth interval, we penalize the maximum bin occupancy:

$$S_{\text{conc}} = \begin{cases} 1, & \text{if } p_{\max} \leq 2/K \\ 1 - \min\left(1, \frac{p_{\max} - 2/K}{0.5 - 2/K}\right), & \text{otherwise} \end{cases} \quad (8)$$

where  $p_{\max} = \max_k(n_k)/N$  is the maximum bin probability. This formulation (Eq. 8) tolerates up to twice the ideal concentration ( $2/K$ ) without penalty, then linearly decreases the score as concentration increases.

**Range Utilization** ( $S_{\text{range}}$ ) . Partition the available depth range into  $K$  equal-width bins and let  $n_k$  be the count in bin  $k$ . Define the number of non-empty bins  $K_+ = \{k \in \{1, \dots, K\} \mid n_k > 0\}$ . The range utilization score is  $S_{\text{range}} = K_+/K$ , which penalizes samples whose depths concentrate within a narrow portion of the range.

The final Depth Distribution Score  $S_{\text{dist}}$  is the weighted sum of these three scores:

$$S_{\text{dist}} = \lambda_1 \cdot S_{\chi^2} + \lambda_2 \cdot S_{\text{conc}} + \lambda_3 \cdot S_{\text{range}}, \quad (9)$$

where we empirically set  $\lambda_1 = 0.5$ ,  $\lambda_2 = 0.3$ , and  $\lambda_3 = 0.2$ .

### A.2.2 GRADIENT CONTINUITY SCORE

In the real world, continuous physical surfaces should have smoothly transitioning depth values, without drastic random fluctuations. However, perhaps due to rendering defects in synthetic data, some sample depth maps exhibit gradient abrupt changes caused by noise on smooth surfaces. If these samples are used for training, the model will learn incorrect depth changes, thus affecting prediction quality.

Inspired by the gradient loss function ((Li et al., 2024b; Yang et al., 2018; Ranftl et al., 2020)), we propose a *gradient continuity score* to assess the noise content of each sample. We first calculate the gradient magnitude  $G(i, j) = \sqrt{(\partial_x D)^2 + (\partial_y D)^2}$ . To distinguish reasonable gradient abrupt changes at normal object edges from those caused by abnormal noise, we define edge pixels as pixels with gradient magnitudes in the top 10%. Within the smooth region, we use the coefficient of variation  $CV = \frac{\sigma_G}{\mu_G}$  to assess gradient consistency:

$$S_{\text{grad}} = \frac{1}{1 + CV}, \quad (10)$$

where  $\mu_G$  and  $\sigma_G$  are the mean and standard deviation of the gradient magnitude in the region, respectively.

### A.2.3 TOTAL SCORE

The depth distribution score and gradient continuity score capture different aspects of sample quality. We combine them into a *Total Score*, defined as  $S_{\text{total}} = (S_{\text{grad}} + S_{\text{dist}})/2$ , to assess the overall quality of each sample for dataset filtering (Eq. 9, 10). It's important to note that our goal is not to provide a particularly precise quality assessment method, but rather to design efficient indicators to quickly filter out samples with quality issues. For example, when performing edge detection, we did not use traditional Canny or Sobel algorithms because the detected edge maps often produce unnecessary artifacts and details. Learning-based methods, on the other hand, predict edges that are always several pixels off from their exact locations (Li et al., 2024b; He et al., 2019; Pu et al., 2022; Su et al., 2021), and their inference time is time-consuming, making them unsuitable for rapid filtering of large datasets.

## A.3 VISUALIZATION OF LOW-QUALITY SAMPLES

Figure 8 provides qualitative examples of low-quality samples from five training datasets. It can be seen that some datasets contain samples with highly uneven depth value distributions, leading to biased supervision. This situation motivates us to use a depth distribution score when evaluating dataset quality.

In addition, Figure 9 shows RGB images, gradient maps, and ground-truth depth examples from the same five datasets. The highlighted areas indicate the presence of severe gradient noise or inconsistent edges, which can negatively impact training stability. These qualitative findings support our quantitative gradient consistency metric.

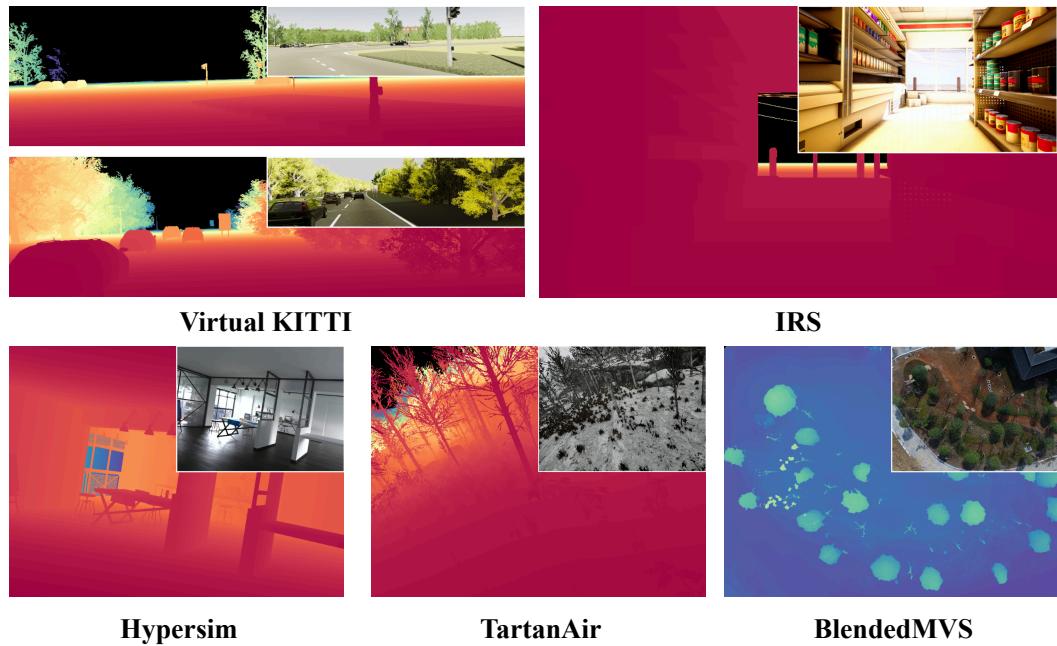


Figure 8: RGB images and GT of each dataset, showing that the depth value distribution of some samples is not uniform.

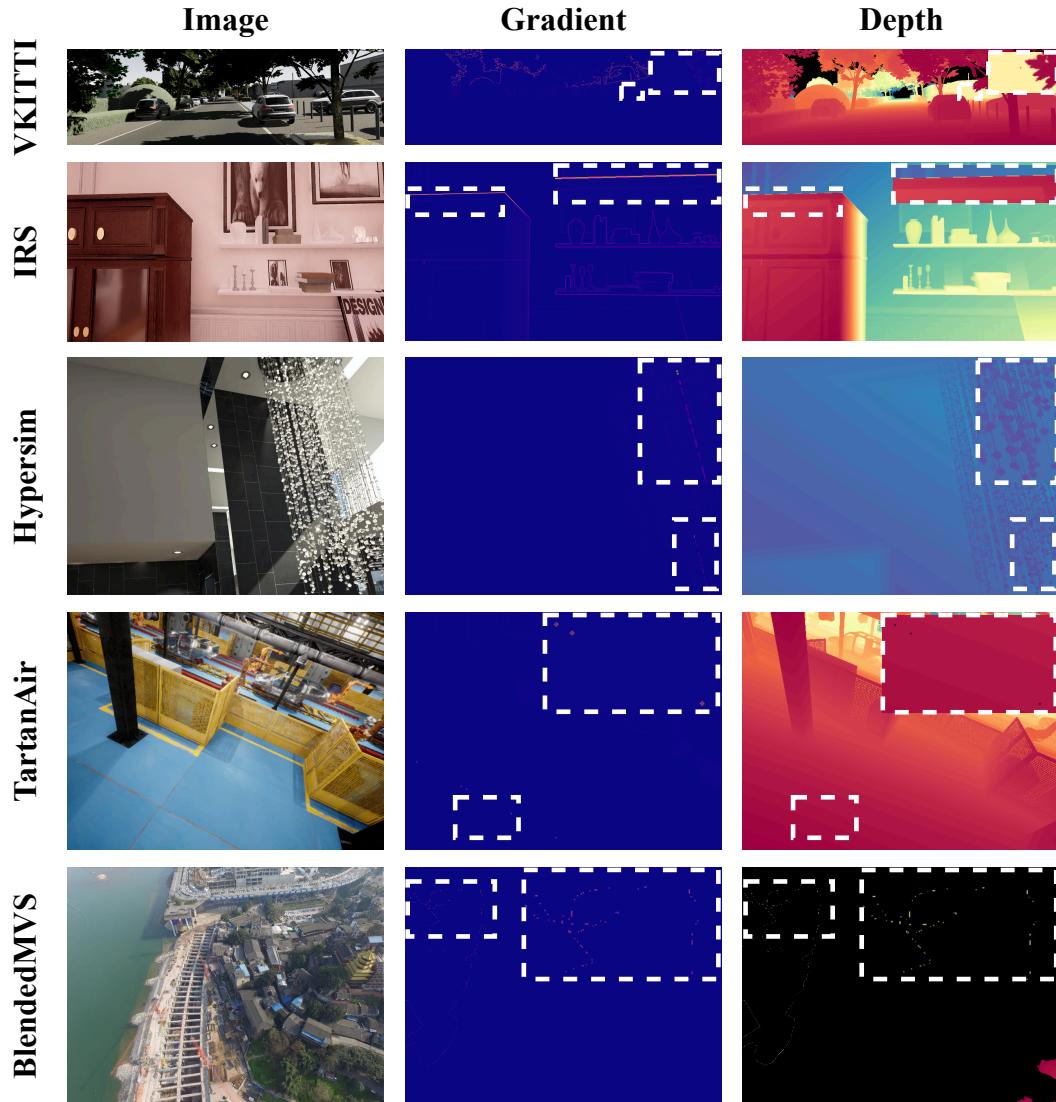


Figure 9: Examples of RGB, gradient, and GT depth from five datasets. The dotted box highlights the noisy area.

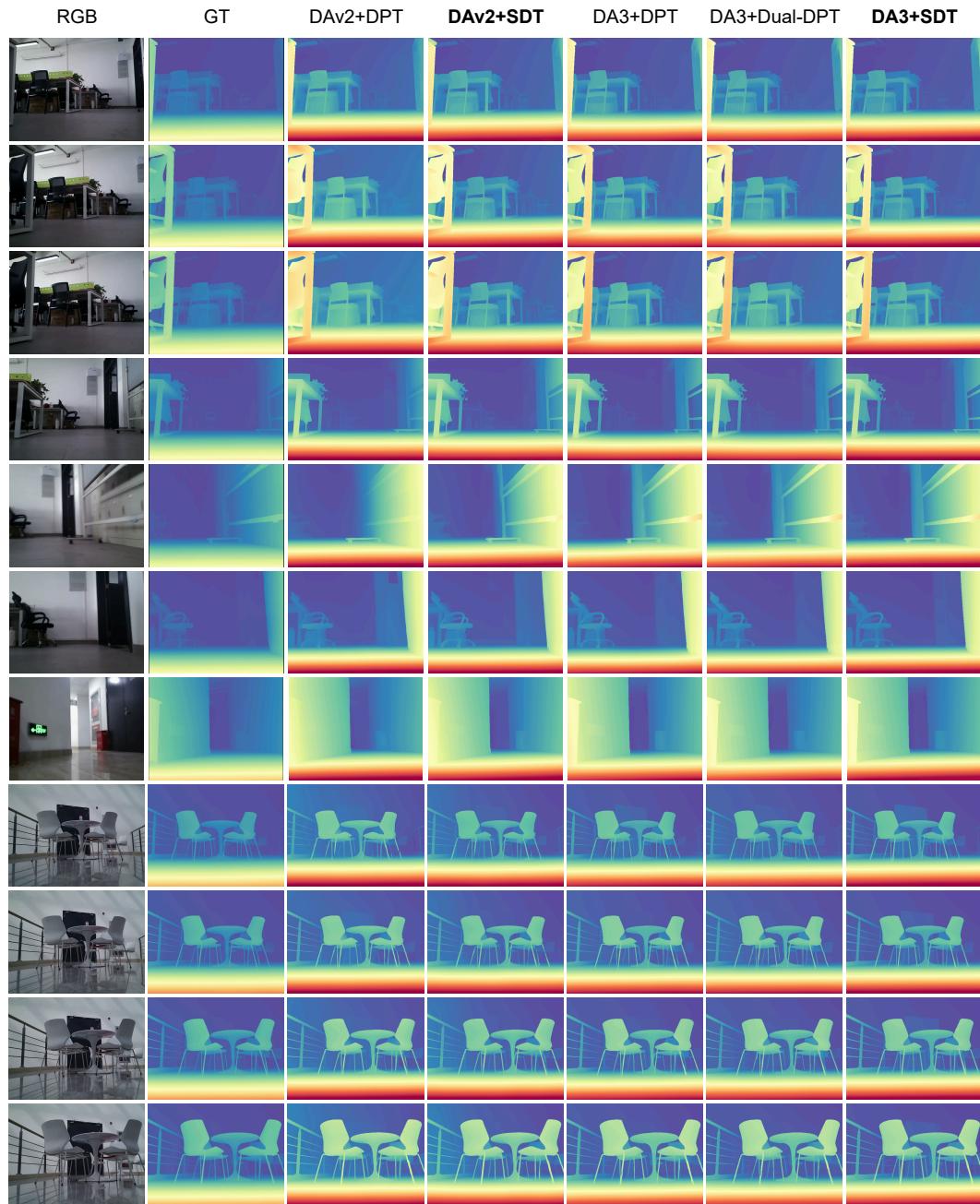


Figure 10: Qualitative results of zero-shot monocular depth estimation with different decoders (DPT, Dual-DPT, and SDT) using the same encoder.