AnyDepth: Efficient Zero-Shot Depth Estimation with Simple Decoding and Data-Quality Filtering

Anonymous authors
Paper under double-blind review

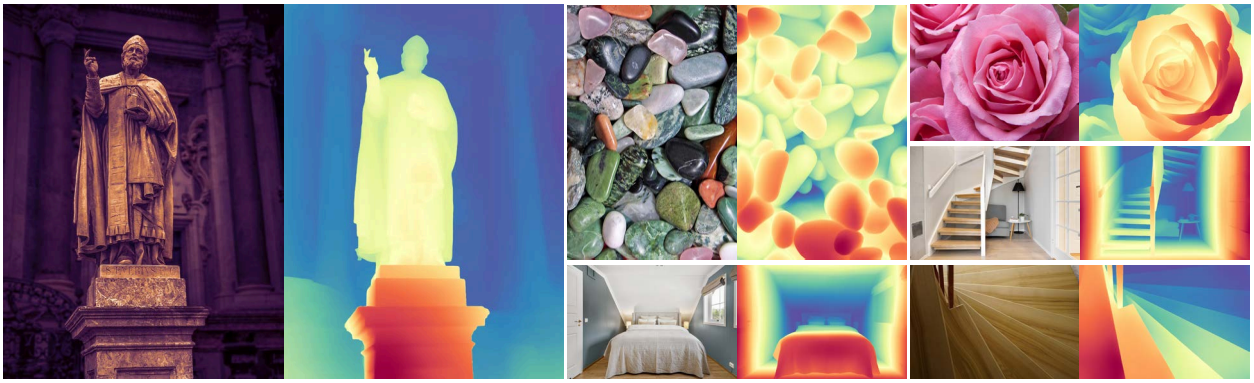
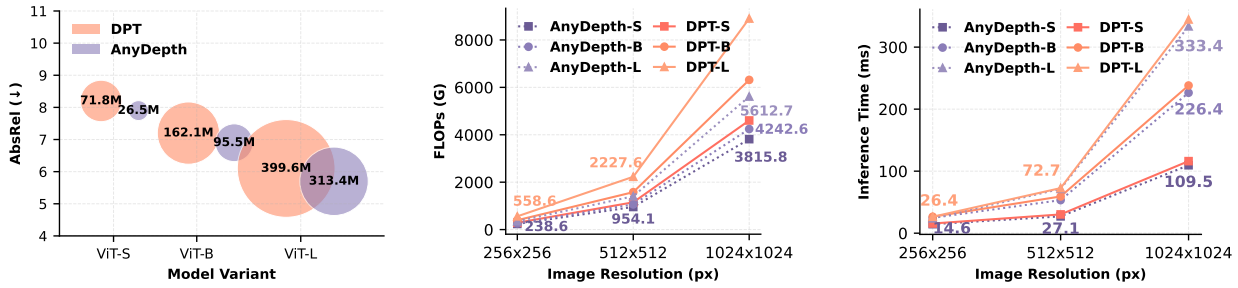


Figure 1: We present **AnyDepth**, an efficient framework for zero-shot monocular depth estimation that studies two controllable sources of efficiency after strong frozen visual features are available: decoder complexity and training-sample quality.

Abstract

Recent monocular depth estimation systems increasingly benefit from strong pretrained visual encoders and large-scale training data. However, once high-quality dense features are available, two remaining sources of cost become especially important: the complexity of Dense Prediction Transformer (DPT)-style multi-branch decoders and the quality of the depth samples used for training. We present AnyDepth, an efficient framework for zero-shot monocular depth estimation that studies these two design choices under controlled settings. AnyDepth uses a frozen DINOv3 encoder and replaces the conventional reassemble-then-fuse DPT head with a Simple Depth Transformer (SDT), a single-path decoder that fuses projected multi-layer tokens before spatial reconstruction. SDT further combines lightweight local refinement with learnable progressive upsampling to improve detail preservation without introducing multi-branch feature alignment. In parallel, we introduce two depth-specific sample quality scores, based on depth distribution and gradient continuity, to filter low-quality training samples before optimization. Across standard indoor, outdoor, synthetic, and robot-scene benchmarks, SDT improves the efficiency-accuracy trade-off relative to DPT under matched encoder and training settings, reducing decoder parameters by 86.6%–89.2% while lowering computational cost and edge-device latency. The filtering strategy reduces the merged training set from 584K to 369K samples and preserves or improves several metrics under controlled comparisons. These results suggest that, in the era of strong frozen visual encoders, decoder simplicity and data quality remain practical control points for reproducible and deployable zero-shot depth estimation. **Website:** `https://anonymous.4open.science/w/AnyDepth-anonymous-site-C61A/`



(a) Model Parameters (b) Computational Cost (FLOPs) (c) Inference Latency

Figure 2: Comparison of model efficiency between **AnyDepth** and DPT. From left to right, we report the number of parameters, computational complexity (FLOPs), and inference latency under different model sizes and input resolutions. AnyDepth consistently achieves lower computational cost and faster inference, particularly at higher resolutions.

1 Introduction

Monocular depth estimation (MDE) aims to infer dense scene geometry from a single RGB image. It is useful for 3D reconstruction (Wang et al., 2025c)(Wang et al., 2025d)(Wang et al., 2025b), image 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 perception (Wu et al., 2025)(Huang et al., 2025a)(Liu et al., 2025b)(Liu et al., 2025a)(Huang et al., 2025b)(Song et al., 2025)(Ye et al., 2025)(Huang et al., 2025c)(Huang et al., 2025d), where depth provides geometric structure without requiring a dedicated depth sensor. Recent progress has substantially improved cross-dataset generalization, especially through pretrained visual transformers, large-scale training data, and foundation-model-style depth systems.

This progress also changes the design question. Earlier robust MDE systems such as MiDaS (Ranftl et al., 2020)(Birkl et al., 2023) and DPT (Ranftl et al., 2021) showed that transformer features and multi-scale decoding improve transfer. The Depth Anything series (Yang et al., 2024a)(Yang et al., 2024b)(Lin et al., 2025) further demonstrates the value of large-scale unlabeled or pseudo-labeled data. These approaches show the power of scale, but they also make two remaining costs more visible. First, DPT-style decoders reassemble tokens from multiple Transformer layers into scale-specific feature maps and then fuse them through a multi-branch structure, introducing repeated spatial reconstruction, cross-scale alignment, parameters, FLOPs, and latency. Second, indiscriminately scaling depth data can introduce samples with concentrated depth ranges, invalid supervision, or noisy depth gradients, weakening training quality and reproducibility.

This work studies the decoder and data pipeline after a strong frozen encoder has already provided high-quality dense features. We propose AnyDepth, a lightweight and data-aware framework for zero-shot MDE. AnyDepth uses a frozen DINOv3 encoder and a Simple Depth Transformer (SDT) decoder. The key architectural change is to fuse multi-layer tokens before spatial reassembly, reversing the common reassemble-then-fuse pattern of DPT. To improve the data pipeline, we score training samples using depth distribution and gradient continuity, then remove low-quality samples before training.

We evaluate these design choices through controlled efficiency-accuracy comparisons. Under matched encoder and training settings, SDT reduces parameters and computation relative to DPT while preserving or improving depth accuracy on multiple zero-shot benchmarks. On a Jetson Orin Nano, SDT also reduces latency and memory relative to DPT, indicating that the decoder simplification has practical deployment consequences. Our contributions are:

- We frame efficient zero-shot MDE as a controlled design problem in which strong frozen features expose decoder complexity and data quality as remaining efficiency bottlenecks.
- We propose SDT, a single-path decoder that performs token fusion before spatial reconstruction and avoids repeated DPT-style cross-scale reassembly.

- We introduce a lightweight sample filtering strategy based on depth distribution and gradient continuity, reducing the training set from 584K to 369K samples while retaining useful supervision.
- We provide matched decoder comparisons, cross-encoder transfer experiments, boundary evaluations, ablations, and edge-device measurements to support the efficiency-accuracy trade-off.

2 Related Work

Zero-Shot Monocular Depth Estimation. Zero-shot MDE seeks models that transfer across scenes and datasets without per-domain retraining (Chen et al., 2016)(Piccinelli et al., 2024)(Chen et al., 2020)(Yin et al., 2021). Many systems learn affine-invariant depth to recover relative structure while handling scale and shift ambiguity (Ranftl et al., 2020)(Yang et al., 2024a)(Yang et al., 2024b). Early robust systems such as DiverseDepth (Yin et al., 2020), MiDaS (Ranftl et al., 2020)(Birkl et al., 2023), DPT (Ranftl et al., 2021), Omnidata (Eftekhar et al., 2021), DepthFormer (Li et al., 2023), and ZoeDepth (Bhat et al., 2023) established dataset mixing, transformer features, and transfer evaluation as central ingredients. More recent methods scale this direction through large data or stronger priors: Depth Anything (Yang et al., 2024a)(Yang et al., 2024b)(Lin et al., 2025) emphasizes broad visual coverage, while Marigold (Ke et al., 2024), GeoWizard (Fu et al., 2024), Lotus (He et al., 2024), and GenPercept (Xu et al., 2024) adapt diffusion or dense-perception priors. These methods demonstrate that scale and foundation priors are powerful. AnyDepth is complementary: it asks what decoder and data choices remain necessary once strong frozen features are already available.

Despite these advances, many existing systems keep the decoder design fixed or only slightly modified while improving the encoder or expanding the training set. This motivates a closer examination of the decoder and the training sample pool under matched encoder settings.

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), local planar guidance (Lee et al., 2019), Swin-Depth (Cheng et al., 2021), PVT (Wang et al., 2021), Uformer (Wang et al., 2022), and DPT (Ranftl et al., 2021) all reflect the value of combining information across resolutions. DPT remains a natural baseline for transformer dense prediction because it first reassembles tokens from several layers into feature maps and then fuses those maps through a cascade. Dual-DPT (Lin et al., 2025) further uses two fusion heads for joint depth and ray prediction. Such designs are effective, but repeated reassembly and cross-scale alignment add overhead, especially at high resolution. SDT tests whether a frozen strong encoder permits a simpler fusion-before-reassembly decoder.

Data quality and deployability. Data-centric learning emphasizes that performance depends not only on architecture but also on data quality, coverage, and noise (Singh, 2023)(Zha et al., 2025). In MDE, some training samples contain concentrated depth ranges or noisy depth gradients, increasing dataset size without necessarily improving useful supervision. AnyDepth therefore uses lightweight sample-quality indicators to filter obvious low-quality depth samples. Efficient depth estimation also matters for resource-constrained platforms. Our Jetson Orin Nano evaluation tests this practical dimension by measuring latency, throughput, and memory under matched decoder settings.

3 The Proposed Method

3.1 Overview

AnyDepth follows a simple design principle: strong frozen visual features should reduce the amount of trainable decoder complexity needed for zero-shot depth estimation. As shown in Fig. 3, given an input image I, we use a pre-trained DINOv3 (Siméoni et al., 2025) encoder to extract multi-level representations from four intermediate Transformer layers T^1, T^2, T^3, T^4. These tokens are linearly projected onto a common dimension, fused before spatial reassembly, reshaped into feature maps, refined by a Spatial Detail Enhancer

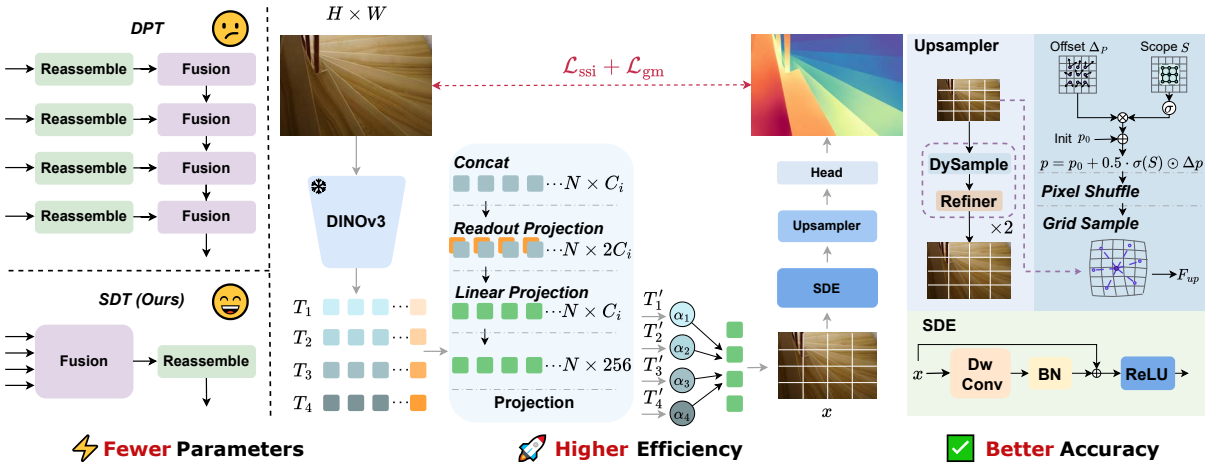


Figure 3: **AnyDepth architecture overview.** The input image is encoded into tokens by a frozen DINOv3 backbone network, then decoded by our lightweight SDT decoder. Tokens undergo only a single projection and weighted fusion. The Spatial Detail Enhancer (SDE) module ensures finer-grained predictions. The feature map is upsampled by an efficient and learnable upsampler DySample, and the depth is finally output by the head.

(SDE), and finally upsampled through a learnable upsampler and prediction head. Unlike DPT (Ranftl et al., 2021), which reassembles each layer and then performs multi-branch cross-scale fusion, SDT performs token fusion first and decodes the result through a single path.

3.2 Simple Depth Transformer (SDT)

Our decoder adopts a single-path fusion and reconstruction strategy designed to use the high-resolution features of DINOv3 without repeatedly rebuilding scale-specific feature maps. We first project the tokens extracted from the encoder into a 256-dimensional space using a linear layer followed by a GELU nonlinearity (Hendrycks & Gimpel, 2016), which preserves informative content while reducing the computational overhead of 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 the weights using a softmax function. This strategy lets the decoder adaptively balance low-level structural details with high-level semantic information before spatial reassembly.

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

where T_i denotes the token in layer i after projection and contains N_p tokens of dimension D. The coefficient $\alpha_i = \frac{\exp(w_i)}{\sum_{j \in \mathcal{L}} \exp(w_j)}$ is derived from a learnable scalar w_i for layer i, so the decoder can adaptively weight shallow and deep representations.

Spatial Detail Enhancer. After the fusion block, we reshape the sequence token output into a spatial feature map. Token-level fusion is efficient, but it can weaken local continuity and obscure shallow texture details that are important for dense prediction. We therefore use a lightweight Spatial Detail Enhancer. The SDE can be expressed by Eq. 2,

$$F' = \mathrm{ReLU}\Big(F + \mathrm{BN}\big(\mathrm{DWConv}_{3\times3}(F)\big)\Big). \tag{2}$$

We implement this operation first using a 3×3 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 $\mathcal{R}(\cdot)$:

$$\mathcal{B}(X) = \mathrm{ReLU}\Big(\mathrm{BN}\big(\mathrm{Conv}_{3\times3}(\mathrm{DySample}_{\times2}(X))\big)\Big), \tag{3}$$

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

$$\mathcal{R}(X) = \mathrm{ReLU}\Big(\mathrm{BN}\big(\mathrm{Conv}_{3\times3}(X)\big)\Big). \tag{5}$$

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

$$\mathcal{U}(X) = \mathcal{R}\Big(\mathcal{S}\big(\mathcal{R}(\mathcal{S}(X))\big)\Big), \tag{6}$$

In this way, the compact feature map of size $H/16\times W/16$ can be progressively upsampled back to the original resolution $H \times W$. The decoder decomposes this process into two $\times4$ upsamplers, using four DySamples of scale 2. Single-stage $\times16$ 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

As shown in Fig. 3, 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.

3.4 Data-Centric Sample Filtering

When training with multiple depth datasets, noise and scale ambiguity can affect optimization, especially in dense prediction. Inspired by data-centric learning (Singh, 2023)(Zha et al., 2025), we treat sample quality as a controllable part of the training pipeline. For MDE, we use two practical indicators: (i) depth values should cover the available range rather than concentrating in a narrow interval, and (ii) gradient magnitudes should be smooth on continuous surfaces while changing near object boundaries. Based on these properties, we define a Depth Distribution Score and a Gradient Continuity Score as fast heuristics for filtering obvious low-quality supervision.

3.4.1 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. 9 , 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_{χ^2}). We measure the deviation from a uniform distribution using the chi-square statistic:

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

where n_k is the number of valid pixels whose depth falls into bin 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_{conc}). To prevent excessive concentration in any single depth interval, we penalize the maximum bin occupancy:

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

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_{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 $K_+ = \{\, k \in \{1, \ldots, K\} \mid n_k > 0 \,\}$. 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_{dist} is the weighted sum of these three scores:

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

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

3.4.2 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 $\text{CV} = \frac{\sigma_G}{\mu_G}$ to assess gradient consistency:

$$S_{\text{grad}} = \frac{1}{1 + \text{CV}}, \tag{10}$$

where μ_G and σ_G are the mean and standard deviation of the gradient magnitude in the region, respectively.

3.4.3 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). The score is designed for efficient filtering of samples with clear distribution or gradient-quality issues. For example, traditional edge detectors often introduce unnecessary artifacts, while learning-based edge detectors add inference cost and may predict slightly shifted edges (Li et al., 2024b)(He et al., 2019)(Pu et al., 2022)(Su et al., 2021), making them less suitable for rapid filtering of large datasets.

3.4.4 Filtering Protocol

After computing the two quality scores, we apply filtering independently within each training dataset. We first remove samples whose valid depth pixels occupy less than 20% of the image, since these samples provide limited dense supervision. For the remaining samples, we sort the depth maps by the Depth Distribution Score and remove the lowest 20%, then sort by the Gradient Continuity Score and remove the lowest 20%. The filtering is performed within each dataset rather than over the merged pool, so that large datasets do not dominate the threshold and smaller datasets retain their scene diversity. The retained samples from all datasets are then merged for training. This procedure reduces the training pool from 584K to 369K samples, with the per-dataset counts reported in Table 1. The filtering step is applied before optimization and does not require an additional learned model.

Table 1: Dataset statistics of good and bad samples.

Dataset	**Total**	**Good**	**Bad**
Hypersim (Roberts et al., 2021)	39,648	26,912	12,736
VKITTI2 (Cabon et al., 2020)	19,559	12,643	6,916
BlendedMVS (Yao et al., 2020)	115,142	74,838	40,304
IRS (Wang et al., 2019)	103,316	68,211	35,105
TartanAir (Wang et al., 2020)	306,637	186,693	119,944
Summary	584,302	369,297	215,005

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., 2020). We only use 369K samples for training. The far plane is set to 100 m. To improve the robustness and generalization of the model, we used data augmentation through flipping and rotation.

Evaluation Datasets and Metrics. We evaluate SDT using eleven datasets containing various scenes: NYUv2 (Silberman et al., 2012), KITTI (Geiger et al., 2013), ETH3D (Schops et al., 2017), ScanNet (Dai et al., 2017), DIODE (Vasiljevic et al., 2019), iBims-1 (Koch et al., 2018; 2020), GSO (Downs et al., 2022), Sintel (Butler et al., 2012), DDAD (Guizilini et al., 2020), Spring (Mehl et al., 2023), and HAMMER (Jung et al., 2023). 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 δ_τ, 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 768×768. 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 ground truth 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 $\mathcal{L}_{\text{ssi}}$ and a gradient matching loss $\mathcal{L}_{\text{gm}}$, and the weight ratio of $\mathcal{L}_{\text{ssi}}$ and $\mathcal{L}_{\text{gm}}$ is set to $1:2$. To stabilize optimization, we follow an optimization strategy similar to

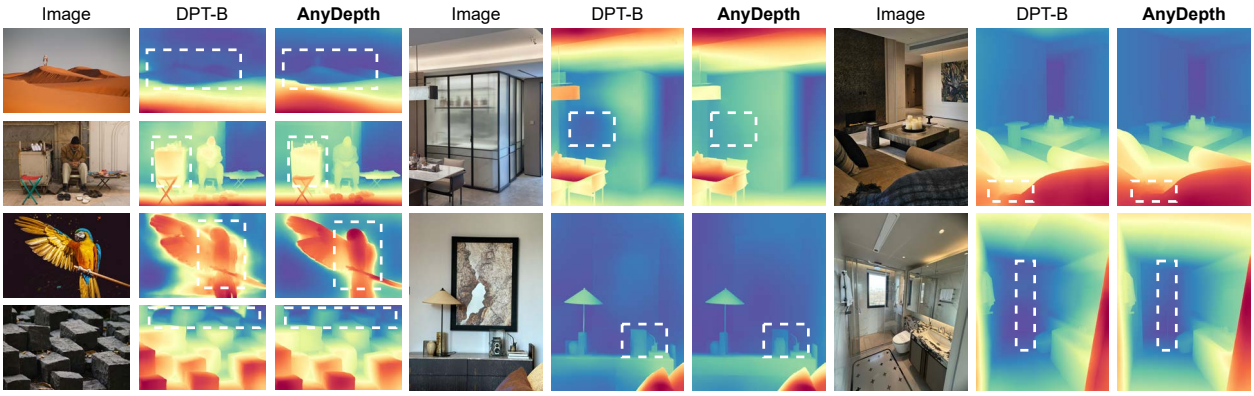


Figure 4: Qualitative results of zero-shot monocular depth estimation using **AnyDepth** with a ViT-B encoder and comparison with DPT-B.

Table 2: **Quantitative comparison of zero-shot affine-invariant depth estimation.** Lower AbsRel values are better; higher δ_1 values are better. DINOv3 (Siméoni et al., 2025) uses the ViT-7B encoder, Depth Anything v2 (DAv2) (Yang et al., 2024b) is trained on 62.6M samples, and Depth Anything 3 (DA3) (Lin et al., 2025) is trained on 22 datasets and over 1276K scenes. For fair comparison, the baseline (DPT (Ranftl et al., 2021)) uses a frozen DINOv3 encoder and DPT head, while our method only 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↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
DINOv3	595K	ViT-7B	7B	4.3	98.0	7.3	96.7	5.4	97.5	4.4	98.1	25.6	82.2
DAv2	62.6M	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
		ViT-L	399.6	4.5	97.9	7.4	94.6	13.1	86.5	–	–	6.6	95.2
DA3	–	ViT-S	34.3	8.5	92.1	10.7	87.0	7.2	93.8	7.7	95.0	24.8	71.9
		ViT-B	135.4	6.3	95.3	10.5	87.4	5.7	96.1	6.8	96.5	23.5	74.4
		ViT-L	410.9	5.0	96.9	7.6	93.5	4.8	96.9	5.5	97.0	22.4	75.0
DPT	584K	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**
		ViT-L	399.6	6.1	**96.8**	8.9	92.5	13.0	94.9	6.0	97.0	23.4	**73.9**
AnyDepth	369K	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**	**8.0**	**94.5**	**6.8**	**95.6**	**23.6**	72.7
		ViT-L	313.4	**6.0**	**96.8**	**8.6**	**92.6**	**9.6**	**95.4**	**5.4**	**97.4**	**22.6**	73.6

DINOv3 (Siméoni et al., 2025). We use AdamW with a base learning rate of 1×10^{-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.

To evaluate the effectiveness of SDT under the fine-tuning of large-scale depth foundation models, we conduct controlled experiments on three representative base models: Depth Anything V2 (Yang et al., 2024b), Depth Anything 3 (Lin et al., 2025), and VGGT (Wang et al., 2025a), while varying only the decoder architecture. For Depth Anything V2 and Depth Anything 3 , we perform full fine-tuning on the Hypersim (Roberts et al., 2021) and Virtual KITTI (Cabon et al., 2020) datasets. For VGGT, due to its significantly larger model size, we adopt parameter-efficient fine-tuning using LoRA (Hu et al., 2022) on the Virtual KITTI (Cabon et al., 2020) dataset, with the LoRA rank set to 8 and the scaling factor α set to 16. All models are trained with an input resolution of 512×512. We use ViT-B as the encoder for DAv2, ViT-L for DA3, and VGGT-1B for VGGT. To ensure a fair comparison, the encoder and decoder learning rates are fixed at 5×10^{-5} and 5×10^{-4}, respectively, across all settings. Each model is optimized using its official loss formulation.

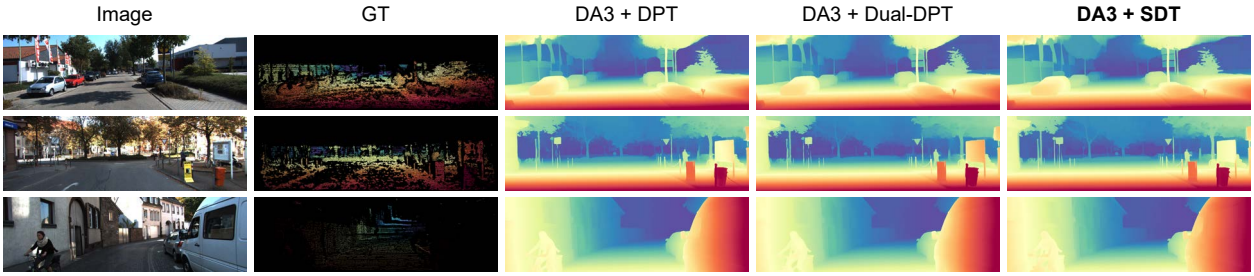


Figure 5: Qualitative results of zero-shot monocular depth estimation using the same DA3 encoder but different decoders on the KITTI dataset. More visualizations are shown in the supplementary material.

Table 3: **Quantitative comparison of zero-shot affine-invariant depth estimation with different encoders and decoders.** Lower AbsRel values indicate better performance, while higher δ_1 values are better. We use ViT-B as the encoder for DAv2, ViT-L for DA3, and VGGT-1B for VGGT. The encoder used pre-trained weights, and the decoder was randomly initialized. Bold numbers denote the better result.

Method	Encoder	Decoder	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
			AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
DAv2	ViT-B	DPT	5.8	96.2	**10.4**	89.1	8.8	94.6	6.2	95.3	**23.4**	73.8
		SDT	**5.6**	**96.4**	10.7	**89.6**	**7.5**	**95.8**	**6.1**	**95.4**	23.9	**73.9**
DA3	ViT-L	DPT	**4.9**	96.9	**8.8**	**92.4**	6.9	95.9	5.0	**96.6**	22.5	74.6
		Dual-DPT	**4.9**	97.0	8.9	**92.4**	7.0	95.8	**4.9**	**96.6**	22.3	74.6
		SDT	**4.9**	**97.1**	8.9	**92.4**	**5.8**	**96.6**	5.0	**96.6**	**21.9**	**74.9**
VGGT	VGGT-1B	DPT	**4.8**	97.7	15.6	77.9	7.2	94.7	**4.6**	97.6	30.7	76.2
		SDT	**4.8**	**98.0**	**15.5**	**80.1**	**7.0**	**95.1**	**4.6**	**98.0**	**30.6**	**76.8**

4.3 Main Results

Quantitative comparisons. Table 2 presents the matched decoder comparison. Since the baselines in the Depth Anything series use DPT-style heads, we compare SDT with DPT under the same frozen DINOv3 encoder settings. Large-scale systems such as Depth Anything (Yang et al., 2024a)(Yang et al., 2024b)(Lin et al., 2025) and DINOv3-7B (Siméoni et al., 2025) provide important context, while our controlled setting isolates the decoder choice. When the encoder and training setup are held fixed, replacing DPT with SDT improves the efficiency-accuracy trade-off. SDT uses only 5–13M decoder parameters across model sizes and substantially reduces the parameter and computational overhead shown in Fig. 2, while maintaining competitive or improved accuracy relative to DPT.

This comparison fixes the backbone, training data, and loss while changing the decoder. It therefore isolates the cost of the decoding head and shows the trade-off targeted by AnyDepth: lower decoder complexity with preserved zero-shot transfer.

Table 3 and Table 4 extend this test to three fine-tuned depth foundation settings with different decoder designs. By fixing the encoder and training protocol, these experiments isolate the effect of decoder choice. Across DAv2, DA3, and VGGT settings, SDT matches or improves over DPT-based decoders on many benchmarks in terms of AbsRel and δ_1, with dataset-dependent variation. Table 5 further reports boundary detection performance under matched settings. SDT achieves higher r_k-F1 scores than DPT-based decoders, supporting the role of local refinement and learnable upsampling in boundary localization.

Across DAv2, DA3, and VGGT-style settings, SDT keeps the comparison focused on the decoder while varying the upstream representations and training recipes. The results show that the same single-path decoder remains effective beyond the frozen DINOv3 setting, with the largest gains appearing in several outdoor and synthetic benchmarks. Table 5 further shows that SDT improves r_k-F1 under matched settings, indicating better boundary localization alongside the global depth metrics.

Table 4: **Quantitative comparison of zero-shot affine-invariant depth estimation on additional benchmarks with different encoders and decoders.** Lower AbsRel values indicate better performance, while higher δ_1 values are better. Experimental settings follow those in Table 3. Bold numbers denote the better result.

Method	Encoder	Decoder	iBims-1		GSO		Sintel		DDAD		Spring		HAMMER	
			AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
DAv2	ViT-B	DPT	4.70	97.57	1.53	99.99	22.91	71.17	13.98	82.25	**27.80**	**65.48**	5.47	98.32
		SDT	**4.37**	**97.92**	**1.50**	**99.99**	**22.18**	**71.45**	**13.45**	**83.38**	29.06	64.14	**5.34**	**98.58**
DA3	ViT-L	DPT	3.46	98.32	1.10	99.99	18.97	78.02	12.20	85.73	19.71	77.90	3.77	99.28
		Dual-DPT	3.43	98.29	**1.07**	99.99	19.09	78.07	12.07	**85.89**	**19.27**	**78.79**	**3.24**	**99.47**
		SDT	**3.38**	**98.36**	1.09	99.99	**18.82**	**78.45**	**12.05**	85.88	19.94	77.87	3.66	99.37

Table 5: Boundary detection performance on each dataset. r_k-F1 denotes the F1 score at radius k pixels. Bold numbers indicate the best result per backbone.

Method	Decoder	iBims-1			Sintel			Spring			HAMMER		
		r_1-F1↑	r_2-F1↑	r_3-F1↑	r_1-F1↑	r_2-F1↑	r_3-F1↑	r_1-F1↑	r_2-F1↑	r_3-F1↑	r_1-F1↑	r_2-F1↑	r_3-F1↑
DAv2	DPT	0.1409	0.2067	0.2763	0.2869	0.3631	0.4266	0.0766	0.1120	0.1498	0.0524	0.0940	0.1389
	SDT (Ours)	**0.1426**	**0.2085**	**0.2789**	**0.3111**	**0.3889**	**0.4528**	**0.0864**	**0.1227**	**0.1615**	**0.0604**	**0.1047**	**0.1506**
DA3	DPT	0.1494	0.2202	0.2952	0.3331	0.4148	0.4811	0.0990	0.1425	0.1902	0.0576	0.1036	0.1520
	Dual-DPT	0.1458	0.2169	0.2923	0.3334	0.4180	0.4862	0.0958	0.1383	0.1847	0.0556	0.1017	0.1508
	SDT (Ours)	**0.1566**	**0.2285**	**0.3042**	**0.3666**	**0.4467**	**0.5098**	**0.1100**	**0.1542**	**0.2016**	**0.0623**	**0.1090**	**0.1577**

Qualitative comparison. Figure 4 compares AnyDepth and DPT-B using the same ViT-B encoder, and Figure 5 compares different decoders under the same DA3 encoder. The visual examples are consistent with the boundary metrics: SDT often produces sharper object boundaries and more coherent local structures.

Efficiency. We evaluate the practical consequences of decoder simplification through parameters, FLOPs, inference latency, and training iteration time. Compared with DPT, SDT reduces the number of parameters (Fig. 2a) and reduces FLOPs by 37% across model sizes, with the largest absolute savings at high resolution (Fig. 2b). It also improves inference speed in the measured settings (Fig. 2c), and the average iteration time of AnyDepth during training is 10% shorter than that of DPT.

The efficiency results span model size, computation, and measured runtime. Parameter count affects the trainable head, FLOPs reflect resolution-dependent computation, and latency tests whether the saving appears in actual inference. The three signals point in the same direction.

4.4 Real World Evaluation

As shown in Fig. 6, 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 onboard computing unit and an Astra Pro RGB-D camera as the perception unit. We evaluate three real-world scenes: a conference room, a corridor, and a rest area. Under the same encoder setup, we compare DPT-style decoders and SDT for perception quality and edge-device efficiency. Table 6 reports inference latency and throughput on the Jetson Orin Nano at two input resolutions. At both 256×256 and 512×512, SDT reduces latency and improves frame rate relative to DPT; at 256 × 256, it also reduces peak memory by approximately 33%. Additional robot-scene depth metrics are reported in Table 10, providing a real-device view of depth quality and runtime cost.

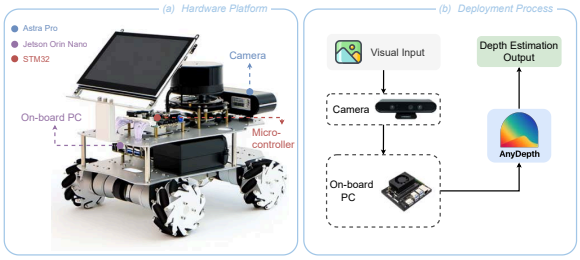


Figure 6: Hardware and evaluation pipeline for real-world experiments.

Table 6: Efficiency comparison of SDT and DPT decoders on a Jetson Orin Nano (4GB) using FP32 precision.

Resolution	**Decoder**	**Params (M)↓**	**Latency (ms)↓**	**FPS↑**	**Peak Memory (MB)↓**
256×256	DPT	99.58	305.65	3.3	589.5
	SDT (Ours)	**13.38**$_{\downarrow 86.56\%}$	**213.35**$_{\downarrow 30.20\%}$	**4.7**$_{\uparrow 42.42\%}$	**395.2**$_{\downarrow 32.96\%}$
512×512	DPT	99.58	1107.64	0.9	–
	SDT (Ours)	**13.38**$_{\downarrow 86.56\%}$	**831.48**$_{\downarrow 24.93\%}$	**1.2**$_{\uparrow 33.33\%}$	–

Table 7: **Ablation experiments of AnyDepth-B on five benchmarks.** We report AbsRel (lower is better) and δ_1 (higher is better).

Components			**NYUv2**		**KITTI**		**ETH3D**		**ScanNet**		**DIODE**	
Filtering	SDE	DySample	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
			9.5	91.1	15.4	77.3	14.0	91.2	8.3	93.5	25.0	71.1
✓			9.3	91.6	15.1	78.1	12.8	90.5	8.0	93.9	24.8	71.1
✓	✓		8.8	92.4	14.7	79.6	11.5	91.0	7.9	94.1	24.3	71.1
✓	✓	✓	**7.2**	**95.0**	**9.7**	**90.1**	**8.0**	**94.5**	**6.8**	**95.6**	**23.6**	**72.7**

4.5 Ablation Study

We conduct ablations to test the two control points in AnyDepth: data quality and decoder design. Table 7 progressively adds filtering, SDE, and DySample to AnyDepth-B. The full system gives the strongest overall results among the tested configurations, supporting the combined role of data filtering, lightweight local refinement, and learnable upsampling. To isolate filtering from the proposed decoder, we additionally train DPT with the original 584K samples and with the filtered 369K samples. Table 8 shows that filtering reduces the training set while preserving competitive DPT performance. Table 9 further validates the effectiveness of each filtering score: using S_{dist} alone improves over no filtering on several metrics, while adding S_{grad} yields the best overall performance across the evaluated datasets. These results indicate that distribution coverage and gradient continuity capture different failure modes in the training data and are both useful for sample selection.

5 Limitations and Future Work

Several limitations remain. First, the filtering strategy is heuristic and provides only modest gains in isolation; a more principled analysis of sample quality, threshold sensitivity, and dataset bias is needed. Second, our Jetson Orin Nano experiments evaluate decoder latency, memory, and robot-scene depth quality, while downstream robotic task metrics such as navigation success, obstacle avoidance, and mapping accuracy remain future work.

6 Conclusion

We presented AnyDepth, an efficient framework for zero-shot monocular depth estimation. Its main lesson is narrow: when the encoder and training protocol are controlled, the decoder and the training samples still matter. SDT reduces DPT-style decoding cost while preserving competitive depth quality, and the filtering scores reduce the training set without relying only on data scale. Across benchmark, ablation, boundary, and edge-device evaluations, decoder simplicity and data quality act as practical levers for the efficiency-accuracy trade-off of zero-shot MDE.

References

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 Birkl, Diana Wofk, and Matthias Müller. Midas v3. 1–a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023.

Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pp. 611–625. Springer, 2012.

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.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560. IEEE, 2022.

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.

Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2485–2494, 2020.

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.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

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.

HyunJun Jung, Patrick Ruhkamp, Guangyao Zhai, Nikolas Brasch, Yitong Li, Yannick Verdie, Jifei Song, Yiren Zhou, Anil Armagan, Slobodan Ilic, et al. On the importance of accurate geometry data for dense 3d vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 780–791, 2023.

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.

Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Korner. Evaluation of cnn-based single-image depth estimation methods. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.

Tobias Koch, Lukas Liebel, Marco Körner, and Friedrich Fraundorfer. Comparison of monocular depth estimation methods using geometrically relevant metrics on the ibims-1 dataset. *Computer Vision and Image Understanding*, 191:102877, 2020.

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.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

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.

Zeting Liu, Zida Yang, Zeyu Zhang, and Hao Tang. Evovla: Self-evolving vision-language-action model. *arXiv preprint arXiv:2511.16166*, 2025b.

Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4981–4991, 2023.

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.

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.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025a.

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*, 2025b.

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*, 2025c.

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*, 2025d.

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/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916. IEEE, 2020.

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.

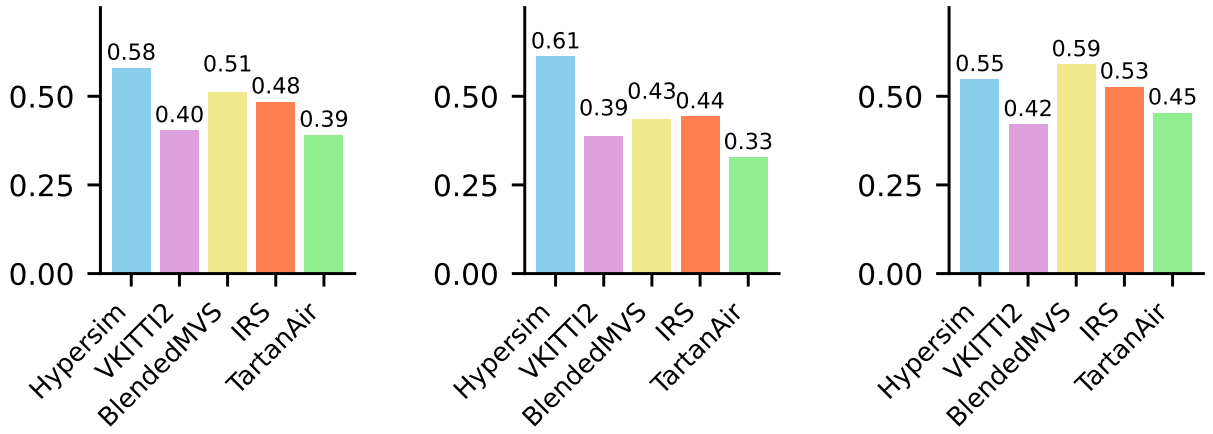
Wei Yin, Xinlong Wang, Chunhua Shen, Yifan Liu, Zhi Tian, Songcen Xu, Changming Sun, and Dou Renyin. Diversedepth: Affine-invariant depth prediction using diverse data. *arXiv preprint arXiv:2002.00569*, 2020.

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) Total Score (b) Depth Distribution (c) Gradient Continuity

Figure 7: Dataset quality evaluated by the total score and two complementary metrics (higher is better).

Results of Data Centric Learning. We apply the two metrics proposed to all training datasets, with the results shown in Fig. 7. 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 Virtual KITTI, 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 Virtual KITTI 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 3.4, 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 per-dataset retained and filtered counts are reported in Table 1. For visualizations of low-quality samples, please see Fig. 9. The merged dataset contains 584K samples, of which approximately 369K are used for training and 215K are filtered out.

Table 8: **Ablation study of data filtering using the DPT decoder.** Lower AbsRel values are better; higher δ_1 values are better. We keep the encoder, decoder, loss, optimizer, input resolution, and training schedule unchanged, and compare DPT trained on the original 584K samples with DPT trained on the filtered 369K samples. Green cells indicate the better result between the two training sets under the same encoder.

Method	**Training Data**↓	**Encoder**	**#Params (M)**↓	**NYUv2**		**KITTI**		**ETH3D**		**ScanNet**		**DIODE**	
				AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
DPT	584K	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**
		ViT-L	399.6	6.1	96.8	**8.9**	**92.5**	13.0	94.9	6.0	97.0	23.4	73.9
DPT	369K	ViT-S	71.8	8.7	92.9	**10.2**	**89.2**	**11.1**	**94.0**	**8.2**	**93.5**	**23.9**	**71.9**
		ViT-B	162.1	**7.3**	**95.5**	**10.6**	**89.0**	**8.8**	**94.0**	**7.0**	**95.7**	**23.6**	73.1
		ViT-L	399.6	**6.1**	**97.1**	9.6	91.4	**10.9**	**95.2**	**5.7**	**97.3**	**22.6**	**74.3**

Table 9: **Ablation study of data filtering criteria using the DPT decoder with ViT-B.** Lower AbsRel values are better; higher δ_1 values are better. We keep the encoder, decoder, loss, optimizer, input resolution, and training schedule unchanged.

Method	Filtering Setting	Training Data↓	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
			AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑	AbsRel↓	δ_1 ↑
DPT	w/o filtering	584K	7.5	95.1	10.8	88.9	10.0	92.9	7.1	95.3	24.5	**73.4**
	w/o S_{grad}	459K	7.3	95.4	**10.4**	**89.2**	10.4	93.3	7.1	95.4	24.9	73.2
	full filtering	369K	**7.3**	**95.5**	10.6	89.0	**8.8**	**94.0**	**7.0**	**95.7**	**23.6**	73.1

Table 8 isolates the effect of filtering under a fixed DPT decoder, encoder, loss, optimizer, input resolution, and training schedule. This comparison supports the main-text claim that filtering can reduce training data while preserving or improving several metrics, but it also shows that the gains are not uniform across all datasets and encoder sizes.

Table 10: Quantitative evaluation on Jetson Orin Nano across three real-world scenarios.

Method	Encoder	Decoder	AbsRel↓	RMSE↓	δ_1 ↑
DAv2 (Yang et al., 2024b)	ViT-B	DPT	24.21	0.1792	82.46
		SDT (Ours)	**23.59**	**0.1693**	**83.11**
DA3 (Lin et al., 2025)	ViT-L	DPT	23.18	0.1722	83.44
		DualDPT	23.08	0.1544	84.55
		SDT (Ours)	**22.66**	**0.1491**	**85.05**

Table 10 reports quantitative depth evaluation on the three robot scenarios discussed in Section 4.4. In addition to AbsRel and δ_1, we report the root mean squared error (RMSE), $\sqrt{\frac{1}{M}\sum_{i=1}^{M}(\hat{d}_i - d_i)^2}$, where M is the number of valid pixels, and $\hat{d}_i$ and d_i denote the predicted and ground-truth depths, respectively. These measurements supplement the main-text Jetson Orin Nano latency and memory results.

A.1 Visualization of low-quality samples

Figure 9 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 10 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.

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 Table 11, SDT consistently and significantly reduces the number of parameters and computational costs across different ViT backbone network sizes and input resolutions.

A.2 Visualization results

Fig. 8 shows the visualization results of our predicted depth and the recovered point cloud. We used the same experimental setup, only replacing the decoder. The results are clearer, more accurate, and the geometry is more obvious and consistent.

Table 11: Decoder efficiency comparison of SDT and DPT heads. Lower Params, FLOPs, and Latency are better. Latency is averaged over 1000 runs on an NVIDIA H100 GPU.

Metric	**Setting**	**DPT**	**SDT (Ours)**
Params (M)↓	ViT-S	50.83	**5.51**$_{\downarrow 89.16\%}$
	ViT-B	76.05	**9.45**$_{\downarrow 87.57\%}$
	ViT-L	99.58	**13.38**$_{\downarrow 86.56\%}$
FLOPs (G)↓	256×256	444.14	**234.17**$_{\downarrow 47.28\%}$
	512×512	1776.56	**936.70**$_{\downarrow 47.28\%}$
	1024×1024	7106.22	**3746.79**$_{\downarrow 47.27\%}$
Latency (ms)↓	256×256	6.66 ± 0.22	$\mathbf{6.10 \pm 0.33}_{\downarrow 8.41\%}$
	512×512	24.65 ± 0.22	$\mathbf{23.17 \pm 0.54}_{\downarrow 6.00\%}$
	1024×1024	99.79 ± 0.79	$\mathbf{93.09 \pm 0.51}_{\downarrow 6.71\%}$

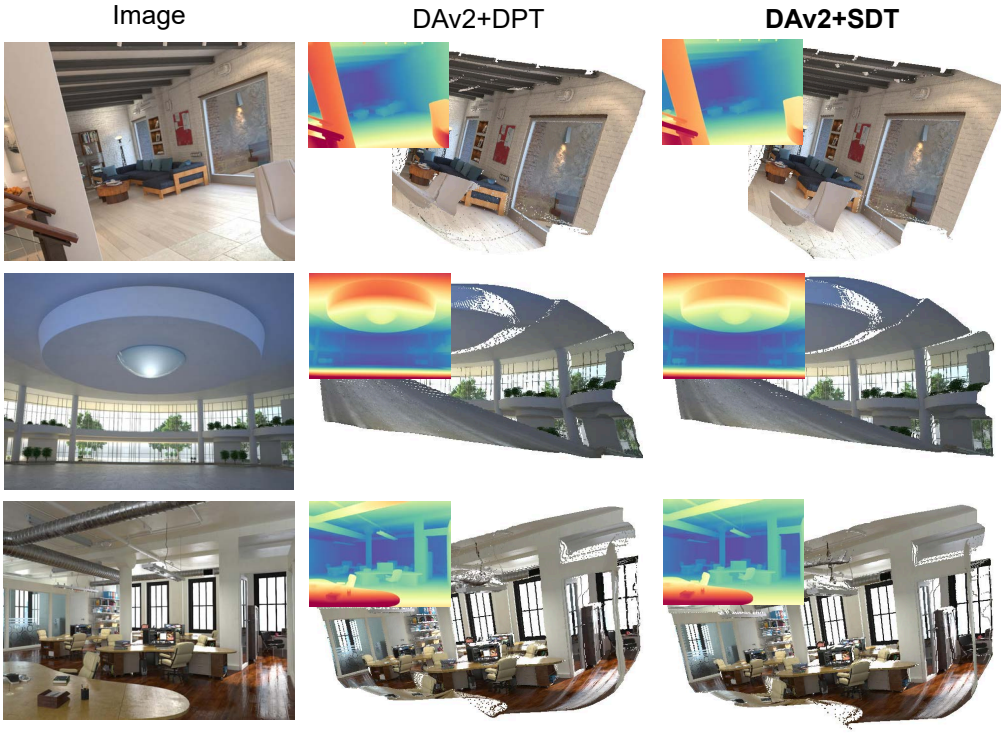


Figure 8: **Point cloud quality comparison.** Point clouds generated using SDT have more regular geometry and lower noise compared to those generated by other methods

Figure 11 shows further qualitative results obtained using the same encoder but different decoders on the KITTI dataset. Compared to other decoders, the depth maps generated by SDT have clearer boundaries and sharper edge structures.

Figure 12 presents qualitative results obtained in real-world scenarios using the same encoder with different decoders. When applied to both DAv2 and DA3, SDT produces depth maps with clearer boundaries and more consistent structures compared to other decoders.

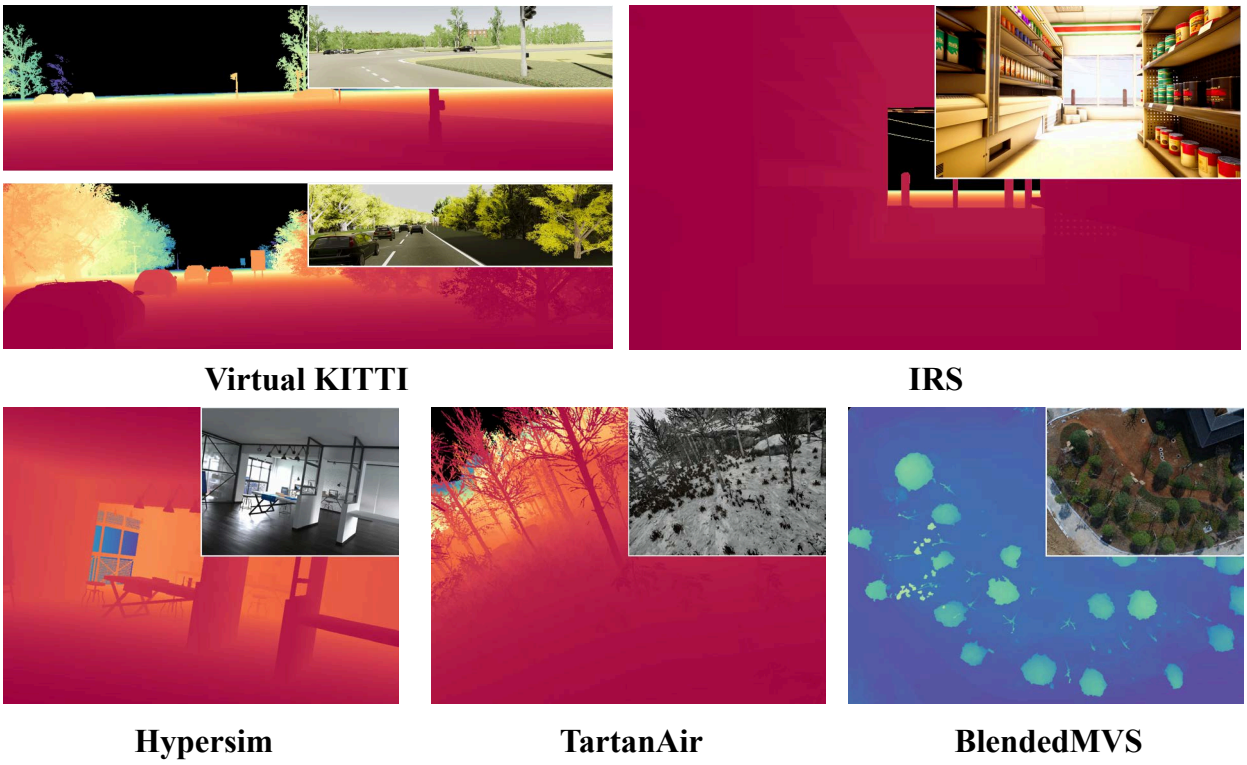


Figure 9: RGB images and GT of each dataset, showing that the depth value distribution of some samples is not uniform.

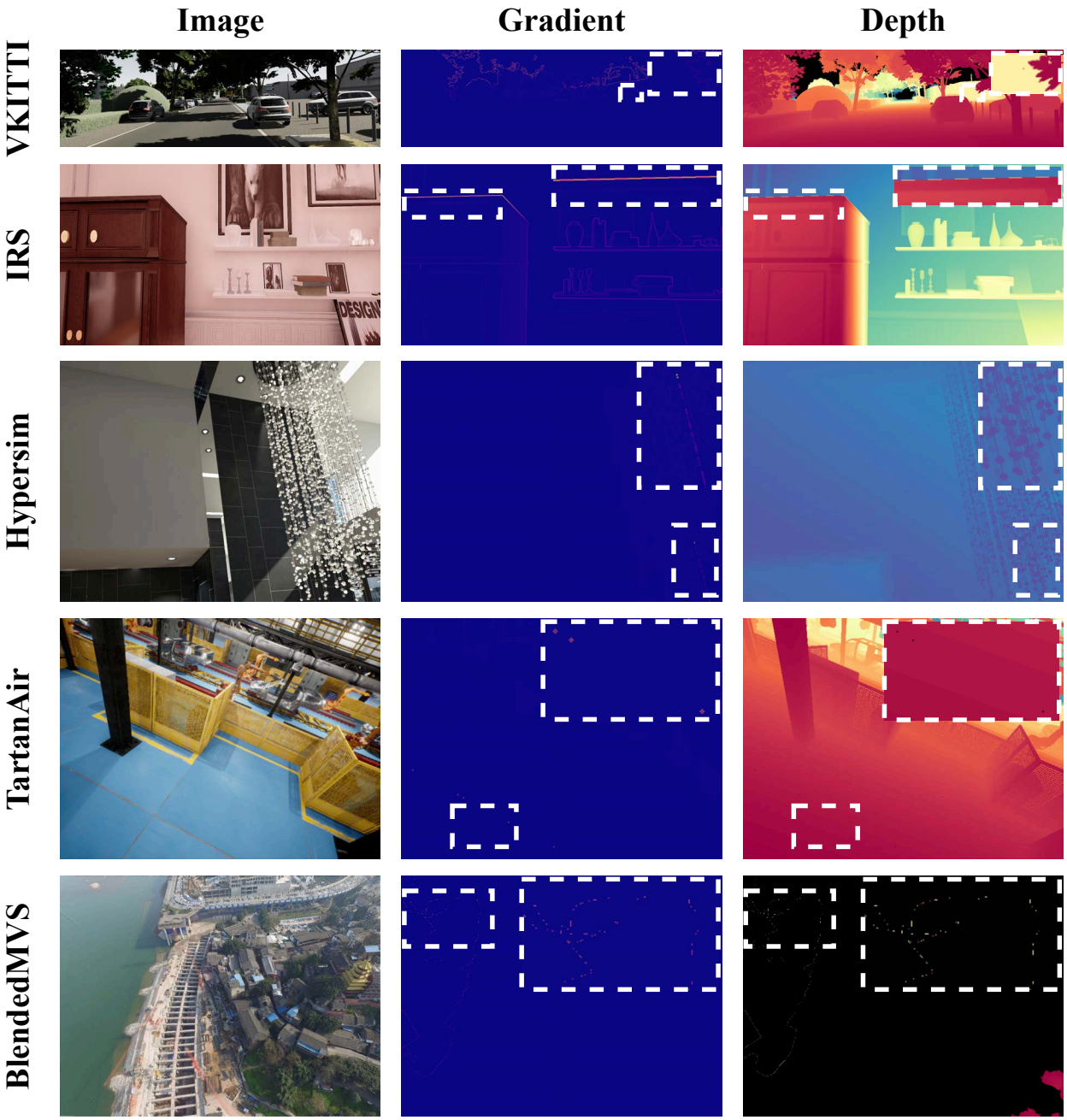


Figure 10: Examples of RGB, gradient, and GT depth from five datasets. The dotted box highlights the noisy area.

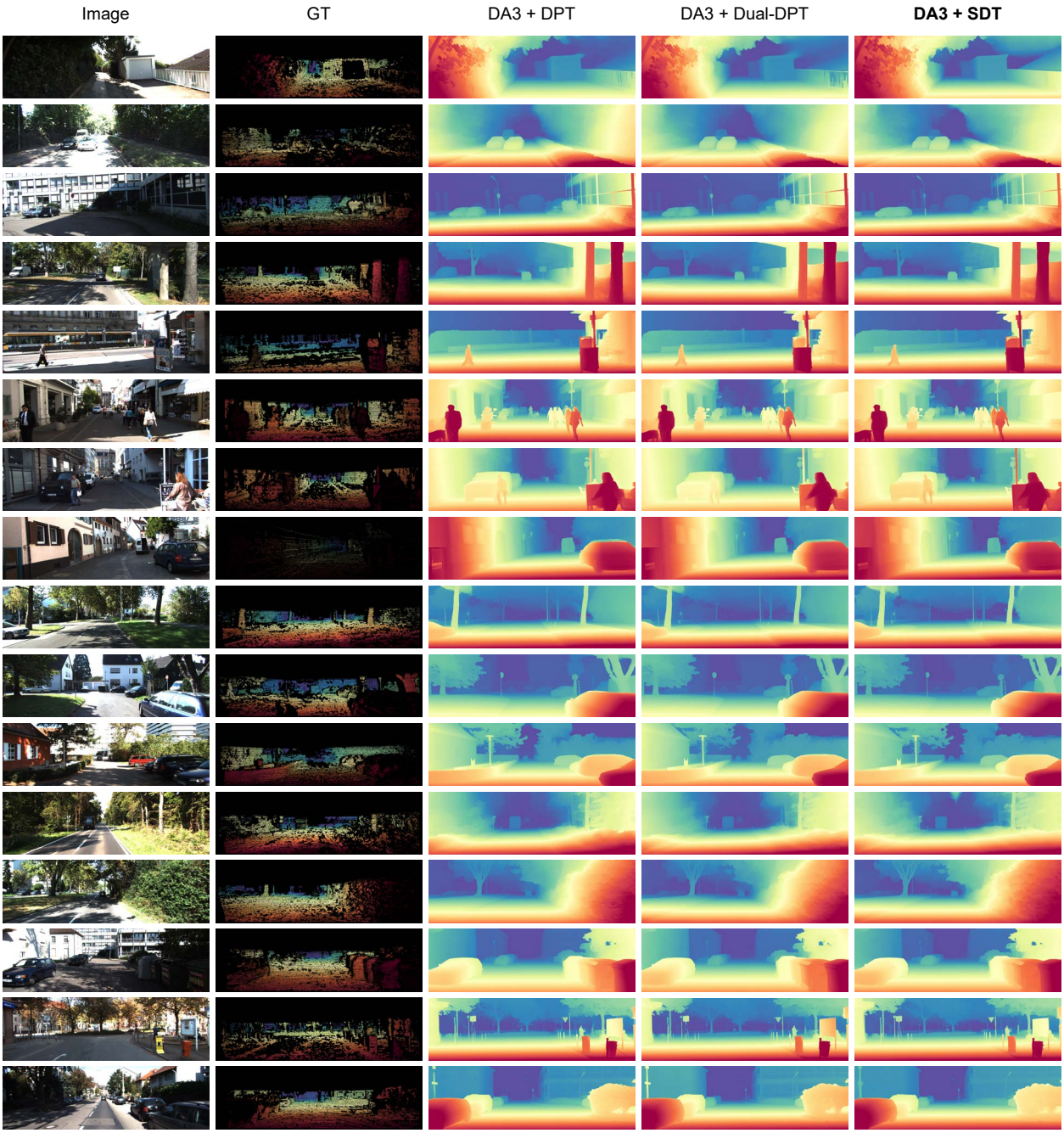


Figure 11: Qualitative results of zero-shot monocular depth estimation using the same DA3 encoder but different decoders on the KITTI dataset.

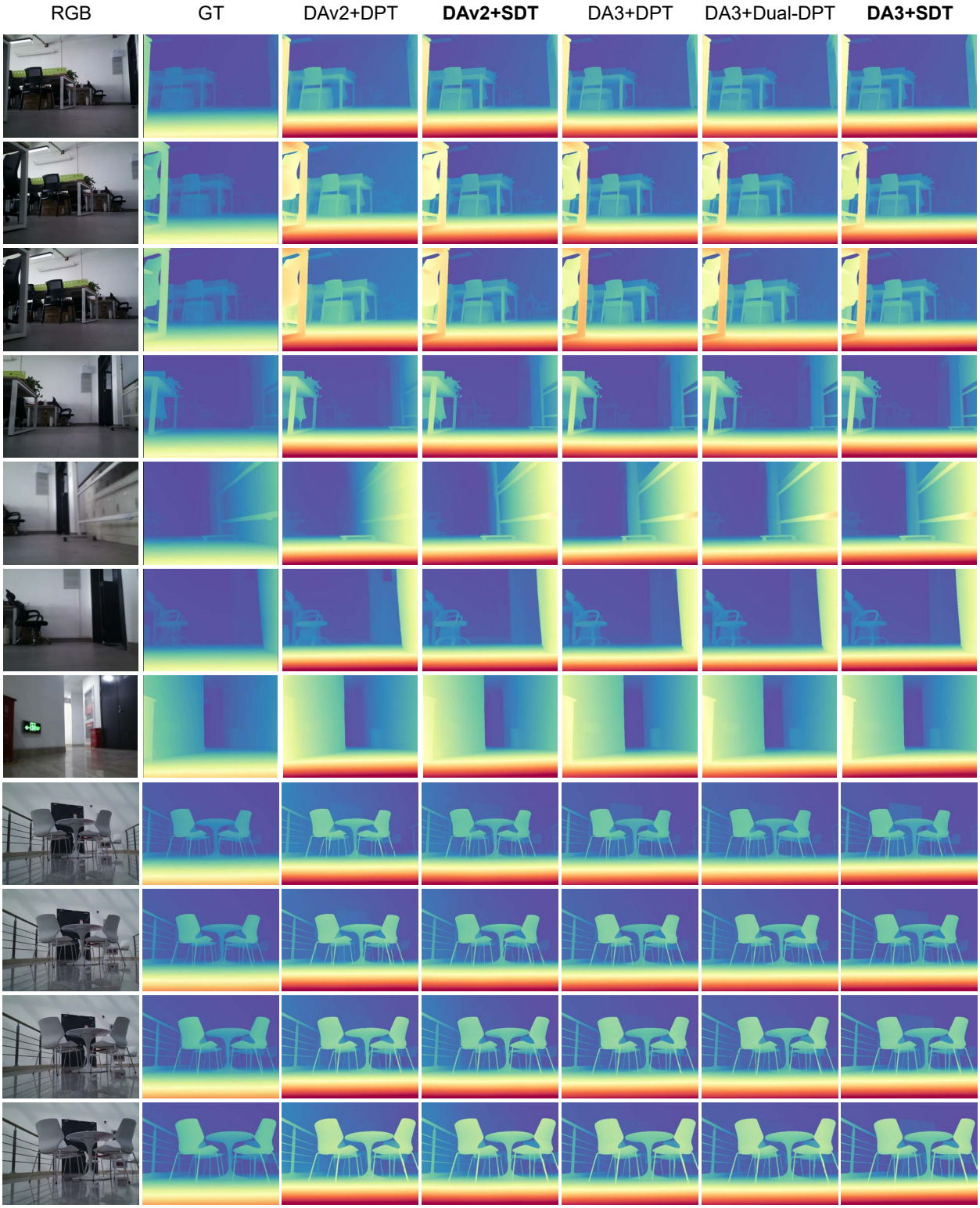


Figure 12: Qualitative results of zero-shot monocular depth estimation were obtained by deploying the same encoder on the robot platform and using different decoders (DPT, Dual-DPT, and SDT).