
MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details

Ruicheng Wang^{1,2*} Sicheng Xu² Yue Dong² Yu Deng² Jianfeng Xiang^{3,2*} Zelong Lv^{1,2*}
Guangzhong Sun¹ Xin Tong² Jiaolong Yang^{2†}
¹USTC ²Microsoft Research ³Tsinghua University

Abstract

We propose MoGe-2, an advanced open-domain geometry estimation model that recovers a metric scale 3D point map of a scene from a single image. Our method builds upon the recent monocular geometry estimation approach, MoGe [61], which predicts affine-invariant point maps with unknown scales. We explore effective strategies to extend MoGe for metric geometry prediction without compromising the relative geometry accuracy provided by the affine-invariant point representation. Additionally, we discover that noise and errors in real data diminish fine-grained detail in the predicted geometry. We address this by developing a unified data refinement approach that filters and completes real data from different sources using sharp synthetic labels, significantly enhancing the granularity of the reconstructed geometry while maintaining the overall accuracy. We train our model on a large corpus of mixed datasets and conducted comprehensive evaluations, demonstrating its superior performance in achieving accurate relative geometry, precise metric scale, and fine-grained detail recovery – capabilities that no previous methods have simultaneously achieved.

1 Introduction

Estimating 3D geometry from a single monocular image is a challenging task with numerous applications in computer vision and beyond. Recent advancements in Monocular Depth Estimation (MDE) and Monocular Geometry Estimation (MGE) have been driven by foundation models trained on large-scale datasets [67, 68, 44, 27, 61, 7]. Compared to depth estimation, MGE approaches often also predict camera intrinsics, allowing pixels to be lifted into 3D space, thus enabling a broader range of applications.

Despite the promising results of recent MGE models, they remain far from perfect and broadly applicable. We expect an ideal MGE method to excel in three key areas: 1) *geometry accuracy*, 2) *metric prediction*, and 3) *geometry granularity*. While accurate global and relative geometry is essential, metric scale is crucial for real-world applications such as SLAM [54, 36], Autonomous Driving [56, 77], and Embodied AI [82, 81, 46]. In addition, recovering fine-grained details and sharp features is also critical for these fields as well as others like image editing and generation [79, 75, 60]. To our knowledge, no existing method addresses all these needs well simultaneously.

In this paper, we introduce a new MGE method towards achieving these goals, while maintaining a simple, principled, and pragmatic design. Our method is built upon the recent MoGe approach [61], which predicts affine-invariant point maps from single images and achieves state-of-the-art geometry accuracy. The cornerstone of MoGe is its optimized training scheme, including a robust and optimal point cloud alignment solver as well as a multi-scale supervision method which enhances local

*Work done during internship at Microsoft Research

†Corresponding author

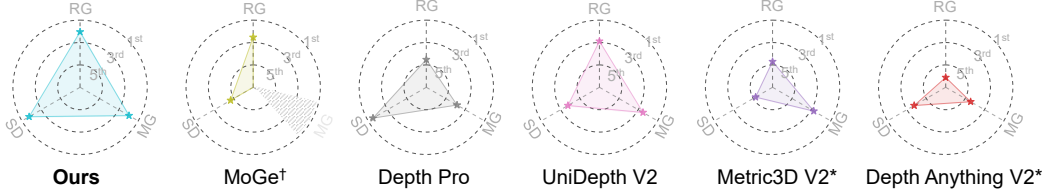


Figure 1: Rankings in comprehensive evaluations. Our method achieves accurate **Relative Geometry** (RG), precise **Metric Geometry** (MG), and **Sharp Detail** recovery (SD) - capabilities not simultaneously achieved by previous approaches. * Methods do not predict camera intrinsics and are evaluated on depth only. † MoGe [61] does not predict metric scale. Please refer to Sec. 4.1 for details.

geometry accuracy. Our work extends MoGe [61] by introducing metric geometry prediction capabilities and improving its geometry granularity to capture intricate details.

For metric geometry estimation, a straightforward solution involves directly predicting absolute point maps in metric space. However, this is suboptimal due to the focal-distance ambiguity issue [61]. To address this, we explore two simple, intuitive, yet effective alternatives. The first uses a shift-invariant point map representation which directly integrates metric scale into point map prediction. The second retains affine-invariant representation but additionally predicts a global scale factor in a decoupled manner. Both strategies mitigate the focal-distance ambiguity, but the latter yields more accurate results, likely due to its well-normalized point map space that better preserves relative geometry.

In the latter regard, we propose a pragmatic data refinement approach to generate sharp depth labels for real-world training data. Real data labels are often noisy and incomplete, particularly at object boundaries, which impede fine geometry detail learning. Previous works such as Depth Anything V2 [68] have opted to use only synthetic data labels, sacrificing the geometry accuracy, despite being sharp upon 2D visualization. Similarly, Depth Pro [7] employs only synthetic data in their second of the two stages. In contrast, we embrace real data throughout the training to ensure high geometry accuracy – a critical goal for our method. Our pipeline filters mismatched or false depth values in real data, primarily found around object boundaries, followed by edge-preserving depth inpainting to fill missing regions using a model trained on synthetic data. This approach results in significantly finer details, with geometry accuracy comparable to models trained on full unprocessed real data.

We train our model on an extensive collection of synthetic and real datasets and conduct a comprehensive evaluation across various datasets and metrics. Experiments demonstrate that our method achieves superior performance in terms of relative geometry accuracy, metric scale precision, and fine-grained detail recovery, surpassing multiple recently proposed baselines, as shown in Fig. 1.

Our contributions are summarized as follows:

- We introduce a Metric MGE framework with the representation of decoupled affine-invariant pointmap and global scale. We provide both insights and empirical evidences for this design.
- We propose a pragmatic real data refinement approach which enables sharp detail prediction while maintaining the generality by fully leveraging large scale real data.
- Our method achieves state-of-the-art results in both geometry accuracy and sharpness, significantly surpassing prior methods in global and local geometry accuracy.

We believe our method enhances monocular geometry estimation’s potential in real-world applications and can serve as a foundational tool facilitating diverse tasks such as 3D world modeling, autonomous systems, and 3D content creation.

2 Related Works

Monocular metric depth estimation. Early works in this field [13, 15, 71, 4, 20] primarily focused on predicting metric depth in specific domains like indoor environments or street views, using limited data from certain RGBD cameras or LiDAR sensors. With the increasing availability of depth data from various sources, recent methods [5, 74, 23, 67, 68, 44, 7] have aimed to predict metric depth in open-domain settings. For example, Metric3D [74, 23] utilized numerous metric depth datasets

and introduced a canonical camera transformation module to address metric ambiguity from diverse data sources. ZoeDepth [5] built on a relative depth estimation framework [47, 6] that is pre-trained on extensive non-metric depth data and employed domain-specific metric heads. UniDepth [44, 45] instead simultaneously learned from metric and non-metric depth data to improve generalizability. Our method focuses on metric geometry estimation and also enables metric depth estimation by directly using the z-channel from the predicted point map, outperforming existing approaches in open-domain metric depth predictions.

Monocular geometry estimation. This task aims to predict the 3D point map of a scene from a single image. Common approaches [72, 73, 44, 45] decouple point map prediction into depth estimation and camera parameter recovery. For instance, LeRes [72] estimates an affine-invariant depth map and camera focal and shift with two separate modules. UniDepth series [44, 45] predicted camera embeddings and facilitate depth map prediction with the estimated camera information. Along another line, DUS3R [62] proposed an end-to-end 3D point map prediction framework for stereo images, bypassing explicit camera prediction. In a similar vein, MoGe [61] predicted an affine-invariant point map for monocular input, achieving state-of-the-art performance with a robust and optimal alignment solver. However, it does not account for metric scale and lacks the finer details, thereby limiting its applicability in many downstream tasks.

Depth prediction with fine-grained details. Numerous methods [40, 35, 45, 68, 27, 7, 25, 39, 80, 66] have been developed to recover fine-grained details in depth prediction. Some [40, 35] enhance local details by fusing depth maps for image patches, but suffer from stitching artifacts. Other works [27, 16, 18] leverage pretrained image diffusion models [50] to generate detailed depth maps. Depth Anything V2 [68] highlights the importance of synthetic data labels by finetuning a DINOv2 [43] encoder with synthetic data and distilling from a larger teacher model. However, synthetic-to-real domain gaps persist and hinder the prediction accuracy. Depth Pro [7] integrates multi-patch vision transformers [11] and a synthetic data training stage, significantly improving depth map sharpness over previous methods, but still falls short in geometric accuracy. In contrast, our model achieves both fine detail recovery and precise geometry through the joint use of synthetic data and real data with a carefully designed real data refinement strategy.

RGB-depth data misalignment artifacts Despite their overall accuracy, depth datasets captured with LiDAR [64, 53, 56, 19] or structure-from-motion (SfM) reconstructions [78, 70, 34] often exhibit various misalignment artifacts. Common issues include spatial misalignment caused by sensor asynchrony [76], ghost surfaces, and incomplete surface reconstruction [70]. Existing methods address LiDAR-specific issues using stereo cues [56] or epipolar geometry [84], while SfM artifacts are mitigated by regenerating depth maps with neural rendering [37, 2]. However, these approaches are often tailored to specific types of artifacts or rely on computationally expensive pipelines. We propose a unified data refinement approach that can handle diverse misalignment artifacts in RGB-depth data regardless of their source or underlying error patterns.

3 Methodology

Our method processes a single image to predict the 3D point map of the scene, achieving accurate relative geometry, metric scale, and fine-grained detail. It builds upon the recent MoGe approach [61] that focuses on affine-invariant point map reconstruction (Sec. 3.1). We explore effective strategies to extend it to accurate metric geometry estimation (Sec. 3.2). Additionally, we develop a data refinement approach that fully leverages real-world training data to achieve both precise and detailed geometry reconstruction simultaneously (Sec. 3.3).

3.1 Preliminaries: MoGe

Given a single image $I \in \mathbb{R}^{H \times W \times 3}$, MoGe estimates an affine-invariant 3D point map $\hat{P} \in \mathbb{R}^{H \times W \times 3}$ with an unknown global scale and shift relative to the ground truth geometry P , achieved by learning through a robust L_1 loss:

$$\mathcal{L}_G = \sum_{i \in \mathcal{M}} \frac{1}{z_i} \|s^* \hat{p}_i + t^* - p_i\|_1, \quad (1)$$

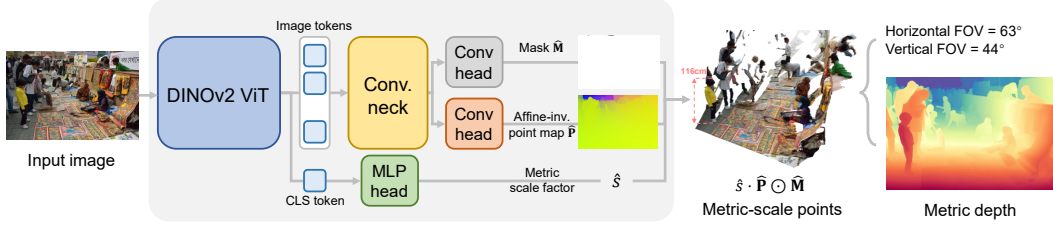


Figure 2: Overview of our model architecture. With the key insight of decoupling metric MGE into affine-invariant point map prediction [61] and global scale recovery, our network design extends MoGe [61] with an additional head for metric scale prediction. This design preserves the benefits of affine-invariant representations for accurate relative geometry while enabling metric scale estimation with the global features captured by the ViT encoder’s classification token.

where $\mathcal{M}$ is the valid mask of ground truth point map, $1/z_i$ is a weighting scalar using inverse ground truth depth, and s^* and t^* are the optimal global scale and shift alignment factors derived by a *robust and optimal (ROE) alignment solver* [61],

$$(s^*, t^*) = \operatorname{argmin}_{s, t} \sum_{i \in \mathcal{M}} \frac{1}{z_i} \|s \hat{\mathbf{p}}_i + \mathbf{t} - \mathbf{p}_i\|_1. \quad (2)$$

To enhance local geometry accuracy, it further applies the robust supervision in Eq. (1) to multi-scale local spherical regions

$$\mathcal{S}_j = \{i \mid \|\mathbf{p}_i - \mathbf{p}_j\| \leq r_j, i \in \mathcal{M}\}, \quad (3)$$

centered at sampled ground truth point $\mathbf{p}_j$ with different radius r_j . After obtaining the affine-invariant point map, the camera’s focal and shift can be recovered by a simple and efficient optimization process (see [61] for more details).

While MoGe accurately predicts relative geometries, it falls short in addressing metric scale and lacks fine-grained details, limiting its broader applications. We explore these challenges and propose effective solutions to achieve accurate metric scale geometry estimation and fine-grained detail reconstruction, as detailed below.

3.2 Metric Scale Geometry Estimation

We explore two alternatives to extend MoGe with metric scale prediction, with corresponding design choices illustrated in Fig. 3.

Shift-invariant geometry prediction. As illustrated in Fig. 3-1, a natural extension of MoGe is to predict a shift-invariant point map by absorbing the metric scale s into the affine point map, while computing only the global shift t via ROE alignment during training and resolving it again at inference time. This design bypasses the focal-distance ambiguity [61] and yields reasonable metric reconstruction results (Tab. 4).

However, due to the large variation in scene scale across open-domain images (*e.g.*, indoors vs. landscapes), the predicted values in shift-invariant space span a wide range. This makes scale learning less stable, and inaccurate scale predictions can produce large gradients that interfere with relative geometry learning (*i.e.*, the middle section of Tab. 4). This motivates our choice to decouple scale estimation from the point map prediction entirely.

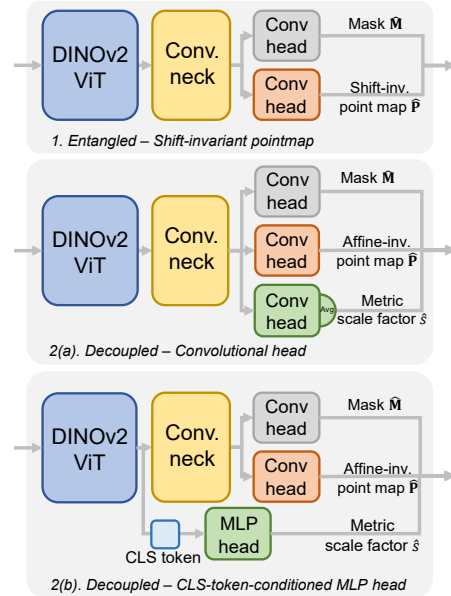


Figure 3: Model design choices for metric scale geometry estimation.

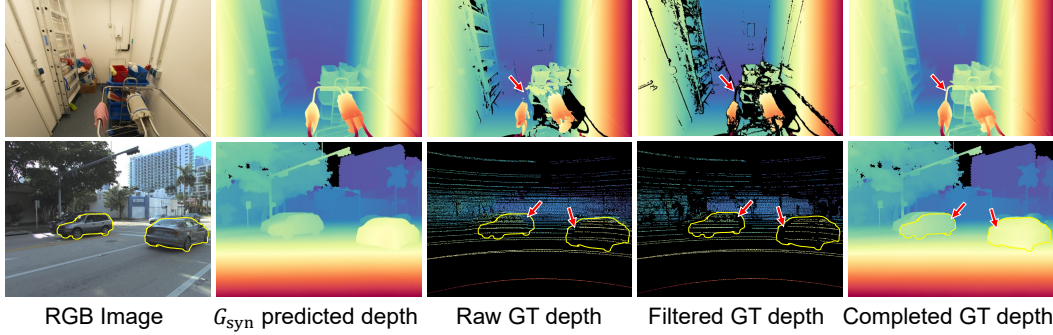


Figure 4: Filtering and completion for real captured datasets. **Top:** The ScanNet++ dataset [70], based on SfM reconstruction, struggles with thin structures and metallic surfaces. Our filtering process removes these artifacts, and our completion scheme reconstructs depth maps that maintain robust absolute depth while compensating for local details that align with the image. **Bottom:** In the Argoverse2 dataset [64], depth and color image discrepancies occur due to temporally unsynchronized sensors. Marking the vehicle boundary in color images (yellow lines) indicates a significant mismatch.

Scale and relative geometry decomposition. To prevent scale affecting relative geometry accuracy, we maintain the geometry branch for affine-invariant point map as in MoGe, and introduce an additional branch for scale prediction with exclusive supervision:

$$\mathcal{L}_s = \|\log(\hat{s}) - \text{stopgrad}(\log(s^*))\|_2^2, \quad (4)$$

where $\log(\hat{s})$ is the predicted metric scale in logarithmic space, and s^* is the optimal scale calculated *online* by Eq. (2) between the predicted affine-invariant point map and the ground truth using the ROE solver. The final metric scale geometry is obtained by multiplying the predicted scale with the affine-invariant point map. We explore two design options for the additional scale prediction branch:

(a) *Convolutional head.* A naive design, as shown in Fig. 3-2(a), is to add a convolution head to output a single scale value, sharing the convolution neck with the affine-invariant point map. However, this approach does not improve relative geometry and worsens metric scale predictions (see Tab. 4). We suspect that simply adding a convolution head results in most information being processed in the convolution neck, which fails to decouple scale prediction from its effect on relative geometry. Moreover, the small output head is ineffective at aggregating local features from the convolution neck, while accurate metric scale prediction requires global information.

(b) *CLS-token-conditioned MLP.* To better decouple relative geometry and metric scale predictions, our second design (Fig. 3-2(b)) uses an MLP head to learn the metric scale directly from the DINOv2 encoder’s classification (CLS) token (see Fig. 2). The global information in the token enables the network to predict an accurate metric scale. As demonstrated in Table 4, such simple design improves metric geometry accuracy compared to the convolution head method while maintaining accurate relative geometry. Thus, we adopt this design as our final configuration.

3.3 Real Data Refinement for Detail Recovery

We found that the MoGe model struggles to accurately reconstruct fine-grained structures due to noise and incompleteness in real training data. Previous studies [68, 27] have also noted this issue and suggest training with synthetic data of sharp labels and pretrained vision foundation models for real-world generalization. However, this still limits geometry accuracy because synthetic data rarely captures real-world diversity. Therefore, using real datasets while reducing their noise and incompleteness is crucial for accurate geometry estimation. To address this, we design a real data refinement pipeline that incorporates synthetic labels to mitigate common failure patterns in real data.

Failure pattern analysis. Real data often originated from LiDAR scans or Structure from Motion (SfM) reconstructions. LiDAR data can suffer from synchronization issues, causing depth and color mismatches, especially at object boundaries. SfM data might miss structures like reflective surfaces, thin structures, and sharp boundaries, as shown in Fig. 4. Our refinement approach leverages the fact that models trained on synthetic data achieve exact color-depth matching and capture sharp, complete

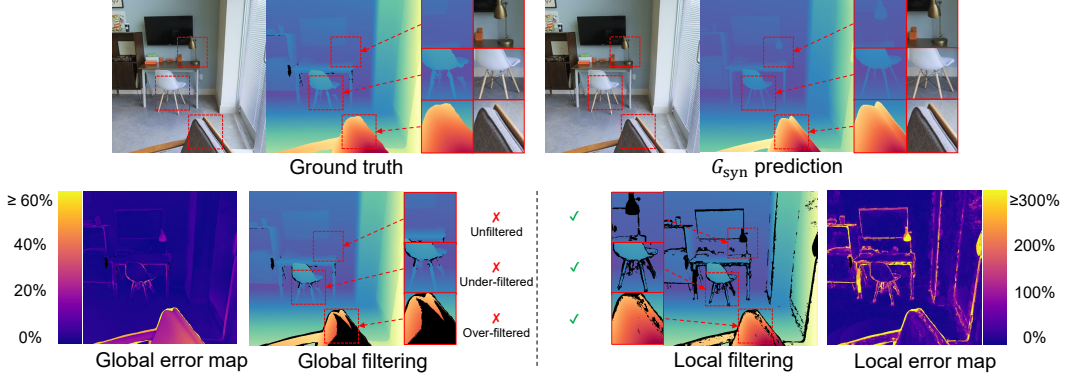


Figure 5: Our mismatch filtering scheme with local geometry alignment effectively avoids depth bias of the predicted results and helps to identify correct artifacts in the real data, whereas a global alignment fails to address the bias and introduces foreground errors, making it unsuitable for filtering.

local geometries. These pseudo labels can help filter incorrect depths and fill in missing parts in real data given accurate local geometries.

Mismatch filtering. To filter real captured depth data, we train a MoGe model solely on synthetic data, denoted as G_{syn} . The model is then applied to real images to infer geometry, which serves as a reference for filtering. While the predicted local structures are generally plausible, G_{syn} often yields biased estimates of the overall scene geometry and layout when used on real-world images. This bias arises from the lack of real-scene priors during training on purely synthetic data. As shown in Fig. 5, such bias can result in incorrect filtering when global errors are considered. Therefore, we focus on comparing the local structures between the real and predicted point maps.

Specifically, given the real-captured points $\{p_i\}$, corresponding predictions $\{\hat{p}_i\}$ by G_{syn} , and the mask $\mathcal{M}$ of valid ground truth, we select a spherical region $\hat{\mathcal{S}}_j$ centered at each estimated point $\hat{p}_j$ with a specific radius $\hat{r}_j$:

$$\hat{\mathcal{S}}_j = \{i \mid \|\hat{p}_i - \hat{p}_j\| \leq \hat{r}_j, i \in \mathcal{M}\}. \quad (5)$$

Within this local region $\hat{\mathcal{S}}_j$, we align the corresponding real-captured points $\{p_i\}_{i \in \hat{\mathcal{S}}_j}$ with the predictions $\{\hat{p}_i\}_{i \in \hat{\mathcal{S}}_j}$ via the ROE solver and mark a real-captured point as an outlier if deviates from the predictions beyond the specified radius, forming a set $\mathcal{O}_j$:

$$\mathcal{O}_j = \{i \mid \|s_j^* p_i + t_j^* - \hat{p}_i\| > \hat{r}_j, i \in \hat{\mathcal{S}}_j\}, \quad (6)$$

with (s_j^*, t_j^*) as optimal alignment factors for local regions. The outlier sets derived from all sampled local regions of different $\hat{r}_j$ are combined and excluded from the mask, yielding the final valid area

$$\mathcal{M}_{\text{filtered}} = \mathcal{M} \setminus \left(\bigcup_j \mathcal{O}_j \right). \quad (7)$$

Regarding the choice of radius $\hat{r}_j$ for each sampled anchor point, we follow the same multi-scale strategy in MoGe’s local loss for capturing context at different spatial scales. Specifically, we set $\hat{r}_j = \alpha \cdot \hat{z}_j \cdot \frac{\sqrt{W^2 + H^2}}{2 \cdot \hat{f}}$, where $\hat{z}_j$ is the depth of $\hat{p}_j$, $\hat{f}$ is the predicted focal length, W and H are image width and height, and $\alpha \in \{1/4, 1/16, 1/64\}$ controls the region’s projected size approximately cover the respective ratio of the image size.

Geometry completion. After filtering out mismatch regions, we create a complete depth map by integrating the detailed structures predicted by G_{syn} with the remaining ground truth depth. Specifically, we reconstruct the depth in the filtered-out regions $\{d_i^c\}_{i \in \mathcal{M}_{\text{filtered}}^c}$ using logarithmic-space Poisson completion:

$$\min \sum_{i \in \mathcal{M}_{\text{filtered}}^c} \|\nabla(\log d_i^c) - \nabla(\log \hat{d}_i)\|^2, \quad \text{s.t.} \quad d_i^c = d_i, \forall i \in \partial \mathcal{M}_{\text{filtered}}^c, \quad (8)$$

Table 1: Quantitative evaluation for *relative geometry*. The numbers are **averaged across the 10 evaluation datasets**. The metrics are visualized with a color gradient from **green** (best) to **red** (worst). Numbers in **gray** cells indicate that some test datasets were used in training. Non-applicable cases are marked with "-". Detailed results on each dataset can be found in *suppl. materials*.

Method	Scale-inv.			Point Affine-inv.			Local			Scale-inv.			Depth Affine-inv.			Affine-inv. (disp)			Avg. Rk.↓
	Rel ^p ↓	δ_1^p ↑	Rk.↓	Rel ^p ↓	δ_1^p ↑	Rk.↓	Rel ^p ↓	δ_1^p ↑	Rk.↓	Rel ^d ↓	δ_1^d ↑	Rk.↓	Rel ^d ↓	δ_1^d ↑	Rk.↓	Rel ^d ↓	δ_1^d ↑	Rk.↓	
ZoeDepth	-	-	-	-	-	-	-	-	-	12.7	83.9	8.75	10.1	88.5	9.09	11.1	88.3	8.78	8.87
DA V1	-	-	-	-	-	-	-	-	-	11.7	85.8	8.22	8.76	90.4	6.91	8.63	92.2	5.62	6.92
DA V2	-	-	-	-	-	-	-	-	-	10.7	87.6	6.80	8.48	90.8	6.15	8.82	91.6	5.42	6.12
Metric3D V2	-	-	-	-	-	-	-	-	-	7.92	91.8	3.39	7.66	92.9	4.53	9.51	89.4	6.17	4.70
MASt3R	14.5	82.1	5.45	11.6	86.0	5.45	8.09	92.2	5.40	11.2	86.5	7.65	9.38	89.1	7.97	11.6	87.8	8.60	6.75
UniDepth V1	13.6	85.0	3.83	10.9	88.1	3.95	9.21	91.0	5.55	10.1	89.1	5.12	8.61	91.0	5.67	9.75	89.9	5.92	5.01
UniDepth V2	11.6	87.7	2.98	8.56	91.9	2.55	6.34	94.9	3.10	8.61	90.8	3.10	6.42	93.9	2.80	7.35	93.0	2.75	2.88
Depth Pro	12.4	87.7	3.83	9.93	89.4	4.30	6.91	94.1	3.55	9.81	89.1	5.33	7.65	92.0	5.05	8.42	91.7	5.08	4.52
MoGe	7.46	94.1	2.14	5.69	95.2	2.14	5.50	95.6	2.05	5.77	94.5	2.72	4.51	96.1	2.94	5.58	95.4	3.17	2.53
Ours	10.8	88.5	2.40	7.98	91.7	2.23	5.33	95.9	1.35	7.35	92.2	2.12	5.62	94.8	2.02	6.66	93.8	2.17	2.05

Table 2: Quantitative evaluation for *metric geometry*. The numbers are **averaged across 7 datasets**.

Method	Point			Depth			Depth (w/ GT Cam)			Avg. Rk.↓
	Rel ^p ↓	δ_1^p ↑	Rk.↓	Rel ^d ↓	δ_1^d ↑	Rk.↓	Rel ^p ↓	δ_1^p ↑	Rk.↓	
ZoeDepth	-	-	-	39.3	49.9	5.90	-	-	-	5.90
DA V1	-	-	-	31.8	54.8	5.50	-	-	-	5.50
DA V2	-	-	-	29.9	56.6	4.43	-	-	-	4.43
Metric3D V2	-	-	-	-	-	-	18.3	73.9	2.75	2.75
MASt3R	26.2	55.3	4.93	49.7	30.3	6.71	-	-	-	5.82
UniDepth V1	12.1	87.2	2.71	23.2	67.5	3.32	21.4	68.6	2.50	2.84
UniDepth V2	10.1	91.9	2.43	21.3	75.3	2.54	18.5	82.6	2.57	2.51
Depth Pro	13.7	81.9	3.29	27.6	54.4	4.36	-	-	-	3.83
Ours	8.19	93.6	1.64	15.7	76.8	2.21	13.6	87.4	2.00	1.95

Table 3: Evaluation of boundary sharpness using F1 scores (↑) in percentages.

Method	iBims-1	HAMMER	Sintel	Spring	Avg. Rk.↓
ZoeDepth	2.47	0.17	2.30	0.43	7.75
DA V1	3.68	0.76	5.64	1.09	6.75
DA V2	13.9	4.74	32.5	6.10	3.75
Metric3D V2	7.36	1.40	25.3	7.23	5.25
MASt3R	1.24	0.05	1.72	0.15	9.50
UniDepth V1	2.35	0.06	0.73	0.17	9.00
UniDepth V2	11.2	4.40	39.7	7.08	3.75
Depth Pro	14.3	5.36	41.6	11.0	1.50
MoGe	11.4	3.89	26.3	8.36	4.67
Ours	17.9	5.42	35.2	8.63	1.75

where $\mathcal{M}_{\text{filtered}}^c$ is the complement area of $\mathcal{M}_{\text{filtered}}$, $\hat{d}_i$ and d_i denote predicted depth by G_{syn} and the real captured depth, respectively. This strategy ensures that the reconstructed depth aligns with the gradient of the predicted depth at local regions while maintaining the ground truth depth as the boundary condition.

Figure 4 illustrates our filtering and completion process. Our method effectively removes mismatched depths from LiDAR scans and fills in missing content in SfM-reconstructed depth maps. The completed depth map retains sharp geometric boundaries that align with the input image while preserving the robust absolute depth from the original map. The refined training data effectively enhances the model’s sharpness and maintains accurate geometry estimation, as shown in Tab. 4.

4 Experiments

Implementation details. We train our model using a combination of 24 datasets with 16 synthetic datasets [10, 58, 49, 59, 42, 33, 14, 24, 83, 51, 63, 21, 1, 65, 55, 73], 3 LiDAR scanned datasets [17, 64, 53], and 5 SfM-reconstructed datasets [3, 78, 70, 34, 69]. We follow MoGe [61] to balance the weights and loss functions among different datasets, and also adopt their approach for image cropping and data augmentation. More details of the training datasets can be found in *suppl. material*.

We use DINOv2-ViT-Large as the backbone for the full model, and DINOv2-ViT-Base model for all ablation studies to ensure efficiency. Our convolutional heads follow MoGe’s design but remove all normalization layers in order to significantly reduce inference latency. The models are trained with initial learning rates of 1×10^{-5} for the ViT backbone and 1×10^{-4} for the neck and heads. The learning rate decays by half every 25K steps. The full model is trained for 120K iterations with 32 NVIDIA A100 GPUs for 120 hours. Ablation models are trained for 100K iterations. Additional implementation details and runtime analysis are provided in the *supplementary material*.

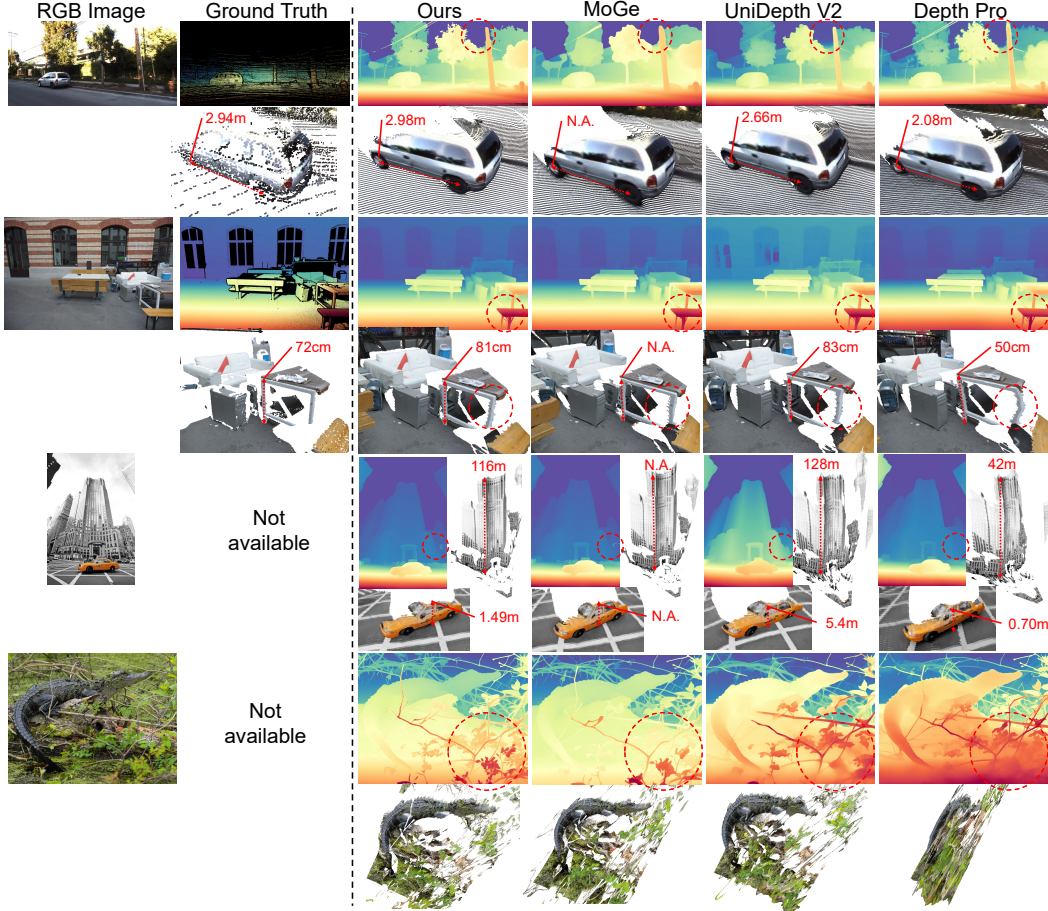


Figure 6: Qualitative comparison of metric scale point and disparity maps. The top two rows are selected from unseen metric scale test datasets. We also labeled key metric measurements in both the ground truth and the estimated geometry. Our estimated metric geometry best matches the ground truth and maintains sharp details. For open-domain inputs, our method produces reasonable geometry with rich details, while results of Depth Pro [7] are severely distorted. *Best viewed zoomed in.*

4.1 Quantitative Evaluation

Benchmarks. We evaluate the accuracy of our method on 10 datasets: NYUv2 [41], KITTI [56], ETH3D [52], iBims-1 [31, 30], GSO [12], Sintel [8], DDAD [19], DIODE [57], Spring [38], and HAMMER [26]. These datasets encompass a wide range of domains, including indoor scenes, street views, object scans, and synthetic animations.

Baselines. We compare our method with several monocular geometry estimation methods, including UniDepth V1 and V2 [44, 45], Depth Pro [7], MoGe [61], MAST3R [32], as well as depth estimation baselines: Depth Anything V1 (DA V1) and V2 (DA V2) [67, 68], ZoeDepth [5] and Metric3D V2 [74]. We evaluate the performance of these methods based on relative scale geometry, metric scale geometry, and boundary sharpness.

Relative geometry and depth. While the primary goal of our method is to estimate metric scale geometry, measuring relative geometry provides valuable insights into how each method reconstructs the geometric shape from the input image. We employ the evaluation metrics of MoGe, measuring over the scale-invariant point maps, affine-invariant point maps, local point maps, scale-invariant depth, affine-invariant depth, and affine-invariant disparity.

Table 1 presents the average relative error – Rel^p ($\|\hat{\mathbf{p}} - \mathbf{p}\|_2 / \|\mathbf{p}\|_2$) for point maps and Rel^d ($|\hat{z} - z|/z$ for depth map), and the percentage of inliers (δ_1^p , where $\|\hat{\mathbf{p}} - \mathbf{p}\|_2 / \|\mathbf{p}\|_2 < 0.25$, and δ_1^d , where

Table 4: Ablation study results **averaged over 10 datasets**, conducted with a ViT-Base encoder.

Configuration	Metric geometry				Relative geometry										Sharpness		
	Point		Depth		Point					Depth							
	Scale-inv.	Affine-inv.	Local	Scale-inv.	Affine-inv.	Affine-inv. (disp)	Rel ^P ↓	δ ^P ↑	Rel ^P ↓	δ ^P ↑	Rel ^P ↓	δ ^P ↑	Rel ^d ↓	δ ^d ↑	Rel ^d ↓	δ ^d ↑	F1 ↑
	Rel ^P ↓	δ ^P ↑	Rel ^d ↓	δ ^d ↑	Rel ^P ↓	δ ^P ↑	Rel ^P ↓	δ ^P ↑	Rel ^P ↓	δ ^P ↑	Rel ^d ↓	δ ^d ↑	Rel ^d ↓	δ ^d ↑	Rel ^d ↓	δ ^d ↑	
Metric scale prediction design																	
Entangled (SI-Log)	10.0	90.7	17.9	68.6	12.9	86.2	10.3	88.8	8.21	93.0	9.83	89.0	7.97	92.0	9.03	91.1	10.7
Entangled (Shift inv.)	8.99	92.1	16.9	68.8	12.0	87.2	9.05	90.2	6.69	94.6	8.46	90.6	6.75	93.2	7.80	92.1	11.8
Decoupled (Conv. head)	9.62	91.4	17.7	68.4	12.2	86.3	9.15	90.0	6.34	94.9	8.46	90.2	6.62	93.2	7.74	92.1	12.7
Decoupled (CLS-MLP)	9.20	91.9	16.5	72.8	11.6	87.6	8.87	90.6	6.26	95.1	8.23	91.0	6.53	93.4	7.53	92.6	12.5
Training data																	
Synthetic data only	12.4	87.3	21.7	65.0	12.3	85.9	9.77	88.9	6.42	94.9	9.04	89.6	7.25	92.5	8.37	91.6	13.3
w/ Raw real data	9.01	92.2	15.8	75.7	11.4	87.8	8.69	90.7	6.37	94.9	8.40	90.4	6.63	93.3	7.69	92.2	10.3
w/ Improved real data	9.20	91.9	16.5	72.8	11.6	87.6	8.87	90.6	6.26	95.1	8.23	91.0	6.53	93.4	7.53	92.6	12.5

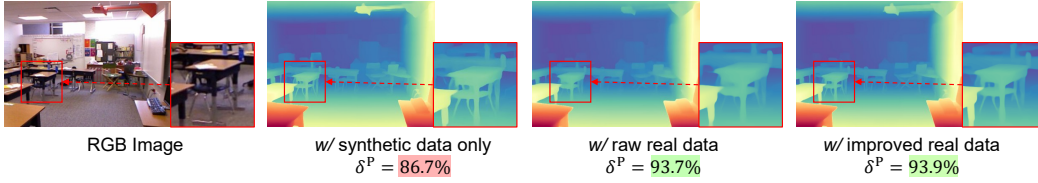


Figure 7: Showcase of ablation study on models trained with different data.

$\max(\hat{d}/d, d/\hat{d}) < 1.25$) across the 10 test datasets, along with the average ranking among the 8 methods. Note that ZoeDepth, DA V1, DA V2, and Metric3D V2 are not evaluated for point settings due to the lack of camera focal prediction. Our method outperforms all existing baselines across every evaluation metric and achieves results comparable to the state-of-the-art relative geometry estimation method, MoGe. This demonstrates that our model *does not compromise the accuracy of relative geometry for achieving metric scale estimation*.

Metric geometry and depth. We evaluate the accuracy of metric scale geometry and depth using 7 datasets with metric scale annotations, including NYUv2 [41], KITTI [56], ETH3D [52], iBims-1 [31, 30], DDAD [19], DIODE [57] and HAMMER [26]. We measure the relative point error (Rel^P) and percentage of inliers (δ_1^P) on estimated metric point maps. Similarly, we evaluate the metric depth accuracy via relative depth error (Rel^d) and depth inliers (δ_1^d). Additionally, we evaluate metric depth estimation using ground truth camera intrinsics for methods that accept this input, which helps eliminate the influence of inaccuracies in the estimated camera intrinsics. As shown in Table 2, our method largely surpasses all existing methods across every metric measurement, demonstrating the advantages of our simple and effective design for decoupling metric scale and affine-invariant point estimation.

Boundary sharpness. To evaluate the sharpness of the estimated geometry, we use two synthetic datasets, Spring [38] and Sintel [8], as well as two real-world test datasets iBims-1 [31] and HAMMER [26], which contain high-quality, densely annotated geometry. We employ the boundary F1 score metric proposed by Depth Pro [7] to measure boundary sharpness. As shown in Table 3, our method achieves boundary sharpness comparable to that of Depth Pro [7] and significantly outperforms it in terms of both relative and metric scale geometry accuracy.

4.2 Qualitative Evaluation

Figure 6 presents a visual comparison of metric scale point maps and disparity maps estimated by different methods. We have annotated key metric scale measurements on both the ground truth and the estimated geometry to facilitate comparison of metric scale accuracy. Our method successfully produces metric scale geometry with sharp details, whereas MoGe and UniDepth V2 miss significant geometric details. Depth Pro exhibits reduced geometric accuracy, particularly in the open-domain test image of a crocodile.

4.3 Ablation Study

Metric scale prediction. In Sec. 3.2, we explored various strategies for accurate metric geometry estimation from open-domain images. We evaluate these configurations across the 10 test datasets using the aforementioned evaluation metrics. We also introduce a naive baseline that directly predicts a metric point map with entangled scale and shift factors using the commonly adopted SI-log loss [13].

Table 4 shows the evaluation results, highlighting the importance of a decoupled design that separates metric scale from relative geometry estimation to improve overall performance. For the scale prediction head, the MLP module outperforms the convolutional head, particularly in metric geometry. This indicates the importance of using global information to predict the metric scale and better decoupling of relative geometry from scale prediction.

Real data refinement. To evaluate the impact of our data refinement pipeline, we conducted ablation study using different data configurations for training – only synthetic data, raw real-world data, and our refined real-world data. As shown in Tab. 4, training exclusively on synthetic data yields the highest sharpness but significantly reduces geometric accuracy. This supports the effectiveness of using synthetic-data-trained model predictions to filter mismatched real data via local error. Training with real-world datasets enhances geometric accuracy but reduces sharpness. Our refined real-world datasets achieve nearly the same geometric accuracy as the original datasets while maintaining reasonable sharpness, as further confirmed by the visual comparison in Figure 7.

5 Conclusion

We have presented MoGe-2, a foundational model for monocular geometry estimation in open-domain images, extending the recent MoGe model to achieve metric scale estimation and fine-grained detail recovery. By decoupling the task into relative geometry recovery and global scale prediction, our method retains the advantages of affine-invariant representations while enabling accurate metric reconstruction. Alongside, we proposed a practical data refinement pipeline that enhances real data with synthetic labels, largely improving geometric granularity without compromising accuracy. MoGe-2 achieves superior performance in accurate geometry, precise metric scale and visual sharpness, advancing the applicability for monocular geometry estimation in real-world applications.

Limitations. Our method struggles with capturing extremely fine structures, such as thin lines and hair, and with maintaining straight and aligned structures under a significant scale difference between the foreground and background. The ambiguity in real-world metric scale can also lead to deviations in out-of-distribution scenarios. We aim to address these challenges by enhancing network architectures and incorporating more real-world priors in the future.

References

- [1] Baidu Apollo. Apollo synthetic dataset, 2019. Accessed: 2025-03-06.
- [2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023.
- [3] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.
- [4] 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*, pages 4009–4018, 2021.
- [5] Shariq Farooq Bhat, Reiner Birkel, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.
- [6] Reiner Birkel, Diana Wofk, and Matthias Müller. Midas v3. 1—a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023.

- [7] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv*, 2024.
- [8] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*, pages 611–625. Springer-Verlag, 2012.
- [9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017.
- [10] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.
- [11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.
- [12] 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, 2022.
- [13] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.
- [14] Michael Fonder and Marc Van Droogenbroeck. Mid-air: A multi-modal dataset for extremely low altitude drone flights. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2019.
- [15] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.
- [16] 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. *arXiv preprint arXiv:2403.12013*, 2024.
- [17] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2D2: Audi Autonomous Driving Dataset. 2020.
- [18] Ming Gui, Johannes S Fischer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Depthfm: Fast monocular depth estimation with flow matching. *arXiv preprint arXiv:2403.13788*, 2024.
- [19] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [20] Vitor Guizilini, Igor Vasiljevic, Dian Chen, Rareş Ambrus, and Adrien Gaidon. Towards zero-shot scale-aware monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9233–9243, 2023.
- [21] Jose L. Gómez, Manuel Silva, Antonio Seoane, Agnès Borrás, Mario Noriega, Germán Ros, Jose A. Iglesias-Guitián, and Antonio M. López. All for one, and one for all: Urbansyn dataset, the third musketeer of synthetic driving scenes, 2023.
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [23] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024.
- [24] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

- [25] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part III 14*, pages 353–369. Springer, 2016.
- [26] 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*, pages 780–791, 2023.
- [27] 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*, pages 9492–9502, 2024.
- [28] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.
- [29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.
- [30] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Körner. Evaluation of cnn-based single-image depth estimation methods. In *Proceedings of the European Conference on Computer Vision Workshops (ECCV-WS)*, pages 331–348. Springer International Publishing, 2019.
- [31] 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 (CVIU)*, 191:102877, 2020.
- [32] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024.
- [33] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3205–3215, 2023.
- [34] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [35] Zhenyu Li, Shariq Farooq Bhat, and Peter Wonka. Patchfusion: An end-to-end tile-based framework for high-resolution monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10016–10025, 2024.
- [36] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos. *arXiv preprint arXiv:2412.04463*, 2024.
- [37] Haotong Lin, Sida Peng, Jingxiao Chen, Songyou Peng, Jiaming Sun, Minghuan Liu, Hujun Bao, Jiashi Feng, Xiaowei Zhou, and Bingyi Kang. Prompting depth anything for 4k resolution accurate metric depth estimation. *arXiv preprint arXiv:2412.14015*, 2024.
- [38] Lukas Mehl, Jenny Schmalfluss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [39] Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Guided depth super-resolution by deep anisotropic diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18237–18246, 2023.
- [40] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9685–9694, 2021.
- [41] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.
- [42] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Transactions on Graphics*, 38(6):184:1–184:15, 2019.

- [43] 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.
- [44] 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 (CVPR)*, 2024.
- [45] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025.
- [46] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.
- [47] 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.
- [48] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.
- [49] 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 *International Conference on Computer Vision (ICCV) 2021*, 2021.
- [50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.
- [51] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [52] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle adjusted direct RGB-D SLAM. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [53] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [54] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.
- [55] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [56] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.
- [57] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, abs/1908.00463, 2019.
- [58] Kaixuan Wang and Shaojie Shen. Flow-motion and depth network for monocular stereo and beyond. *CoRR*, abs/1909.05452, 2019.
- [59] Qiang Wang, Shizhen Zheng, Qingsong Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. IRS: A large synthetic indoor robotics stereo dataset for disparity and surface normal estimation. *CoRR*, abs/1912.09678, 2019.
- [60] Ruicheng Wang, Jianfeng Xiang, Jiaolong Yang, and Xin Tong. Diffusion models are geometry critics: Single image 3d editing using pre-trained diffusion priors. In *European Conference on Computer Vision*, pages 441–458. Springer, 2024.

- [61] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. 2024.
- [62] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024.
- [63] 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. 2020.
- [64] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.
- [65] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *CoRR*, abs/1810.08705, 2018.
- [66] Gangwei Xu, Haotong Lin, Hongcheng Luo, Xianqi Wang, Jingfeng Yao, Lianghai Zhu, Yuechuan Pu, Cheng Chi, Haiyang Sun, Bing Wang, et al. Pixel-perfect depth with semantics-prompted diffusion transformers. *arXiv preprint arXiv:2510.07316*, 2025.
- [67] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.
- [68] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.
- [69] 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. *Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [70] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.
- [71] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5684–5693, 2019.
- [72] 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. *CoRR*, abs/2012.09365, 2020.
- [73] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Simon Chen, Yifan Liu, and Chunhua Shen. Towards accurate reconstruction of 3d scene shape from a single monocular image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6480–6494, 2022.
- [74] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9043–9053, 2023.
- [75] Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. *arXiv preprint arXiv:2406.09394*, 2024.
- [76] Kaicheng Yu, Tang Tao, Hongwei Xie, Zhiwei Lin, Tingting Liang, Bing Wang, Peng Chen, Dayang Hao, Yongtao Wang, and Xiaodan Liang. Benchmarking the robustness of lidar-camera fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3188–3198, 2023.
- [77] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020.
- [78] Amir R Zamir, Alexander Sax, William B Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.
- [79] 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*, pages 3836–3847, 2023.

- [80] Zixiang Zhao, Jianshe Zhang, Shuang Xu, Zudi Lin, and Hanspeter Pfister. Discrete cosine transform network for guided depth map super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5697–5707, 2022.
- [81] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.
- [82] Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13624–13634, 2024.
- [83] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Proceedings of The European Conference on Computer Vision (ECCV)*, 2020.
- [84] Shengjie Zhu, Girish Chandar Ganesan, Abhinav Kumar, and Xiaoming Liu. Replay: Remove projective lidar depthmap artifacts via exploiting epipolar geometry. In *European Conference on Computer Vision*, pages 393–411. Springer, 2024.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#) .

Justification: The main contributions are clearly stated and well supported by the results, with assumptions and scope appropriately reflected in the introduction.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#) .

Justification: The limitations of the work have been discussed in the conclusion section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#) .

Justification: The paper does not include theoretical results or proofs.

Guidelines: The paper does not include theoretical results.

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#) .

Justification: Code and models of this paper will be released and the results will be reproducible.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes] .

Justification: Code and instructions are accessible via an anonymous link in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes] .

Justification: Necessary details are revealed in the main paper and the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes] .

Justification: The experiments are benchmark evaluations with fixed test sets and deterministic metrics, so statistical variation is not applicable.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes] .

Justification: The paper specifies the type of compute used and the approximate runtime per experiment. Additional information on compute requirements is provided in the implementation or appendix section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes] .

Justification: The research complies with the NeurIPS Code of Ethics. All experiments use publicly available datasets with appropriate licenses. No human subjects, sensitive data, or foreseeable misuse risks are involved.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes] .

Justification: The paper outlines potential benefits in applications such as robotics and navigation, and acknowledges general misuse risks associated with depth estimation technologies. While not directly targeting any sensitive domain, the broader impacts are briefly addressed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA] .

Justification: The paper does not release any models or datasets that pose a high risk of misuse. All experiments are conducted on publicly available, well-established benchmarks, and the proposed method does not have known dual-use concerns.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes] .

Justification: All datasets and codebases used in the paper are properly cited with their official papers and licenses. Only publicly released assets with clear terms of use were used. Licenses (e.g., CC BY-NC, GPL, MIT) are respected and referenced in the paper or supplementary material.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#) .

Justification: We release the code and pretrained models along with detailed documentation, including setup instructions, usage examples, and training/evaluation scripts. Anonymized links are provided during the review phase.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#) .

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#) .

Justification: The paper does not involve research with human subjects and therefore does not require IRB approval.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA] .

Justification: LLMs were not used as part of the core methodology or experiments. Any LLM usage was limited to writing assistance and did not impact the scientific content.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Implementation Details

A.1 Network architectures

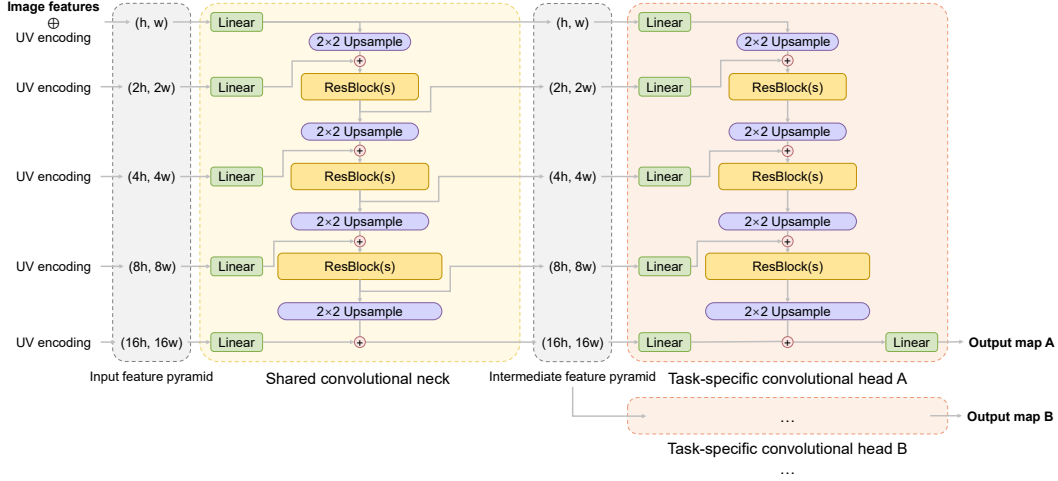


Figure A.1: Illustration of the convolutional neck and head module architectures.

The detailed architectures of our model components are described as follows.

DINOv2 Image Encoder. Our model supports variable input resolutions by leveraging the interpolatable positional embeddings of DINOv2 [43]. The native resolution is determined by a user-specified number of image tokens. Given an input image of arbitrary size and a target number of tokens, we compute a patch-level resolution $h \times w$ that best matches the desired token count. The image is then resized to $(14h, 14w)$ to match DINOv2’s input requirement, and encoded into $h \times w$ image tokens along with one classification token. We extract four intermediate feature layers from DINOv2—specifically, the 6th, 12th, 18th, and final transformer layers—project them to a common dimension, reshape their spatial size to (h, w) , and sum them to form the input for the dense prediction decoder.

Convolutional Neck and Heads. Inspired by prior multi-task dense prediction architectures [48, 28, 61], we design a lightweight decoder consisting of a shared convolutional neck and multiple task-specific heads, as illustrated in Fig A.1. Both the neck and the heads are composed of progressive residual convolution blocks (ResBlocks) [22] interleaved with transpose convolution layers (kernel size 2, stride 2) for progressive upsampling from resolution (h, w) to $(16h, 16w)$. Finally, the output map is resized through bilinear interpolation to match the raw image size. To reduce inference latency on modern GPUs, all normalization layers are simply removed from the ResBlocks, without affecting performance or training stability.

At each scale level of the neck, we inject a UV positional encoding, defined as a mapping of the image’s rectangular domain into a unit circle, preserving the raw aspect ratio information. The resulting intermediate feature pyramid is shared across all heads, each of which independently decodes its respective output map. This design enables multi-scale feature sharing while maintaining head-specific decoding tailored to each prediction task.

CLS-token-conditioned MLP Head. For scalar prediction, we use a two-layer MLP that takes the CLS token feature from DINOv2 as input and outputs a single scale factor, followed by an exponential mapping to ensure a positive scale output. The hidden layer size is equal to the input feature dimension.

A.2 Training Data

The datasets used for training our model are listed in Tab. A.1. All datasets are publicly available for academic use, and their sampling weights follow the protocol established in MoGe [61].

Tab. A.2 provides a rough summary of the number of training frames used by several representative monocular geometry estimation methods. As there is no shared or standardized training set in this field, this table serves to contextualize the scale of training data across methods. Notably, model performance does not necessarily correlate with the amount of training data used.

Table A.1: List of datasets used to train our model.

Name	Domain	#Frames	Type	Weight	Metric Scale
A2D2[17]	Outdoor/Driving	196K	LiDAR	0.8%	✓
Argoverse2[64]	Outdoor/Driving	1.1M	LiDAR	7.1%	✓
ARKitScenes[3]	Indoor	449K	SfM	8.3%	✓
BlendedMVS[69]	In-the-wild	115K	SfM	11.5%	
MegaDepth[34]	Outdoor/In-the-wild	92K	SfM	5.4%	
ScanNet++[9]	Indoor	176K	SfM	4.6%	✓
Taskonomy[78]	Indoor	3.6M	SfM	14.1%	✓
Waymo[53]	Outdoor/Driving	788K	LiDAR	6.2%	✓
ApolloSynthetic[1]	Outdoor/Driving	194K	Synthetic	3.8%	✓
EDEN[73]	Outdoor/Garden	369K	Synthetic	1.2%	
GTA-SfM[58]	Outdoor/In-the-wild	19K	Synthetic	2.7%	✓
Hypersim[49]	Indoor	75K	Synthetic	4.8%	✓
IRS[59]	Indoor	101K	Synthetic	5.4%	✓
KenBurns[42]	In-the-wild	76K	Synthetic	1.5 %	
MatrixCity[33]	Outdoor/Driving	390K	Synthetic	1.3%	✓
MidAir[14]	Outdoor/In-the-wild	423K	Synthetic	3.8%	✓
MVS-Synth[24]	Outdoor/Driving	12K	Synthetic	1.2%	✓
Structured3D[83]	Indoor	77K	Synthetic	4.6%	✓
Synthia[51]	Outdoor/Driving	96K	Synthetic	1.1%	✓
Synscapes[65]	Outdoor/Driving	25K	Synthetic	1.9%	✓
UnrealStereo4K [55]	In-the-wild	8K	Synthetic	1.6	✓
TartanAir[63]	In-the-wild	306K	Synthetic	4.8%	✓
UrbanSyn[21]	Outdoor/Driving	7K	Synthetic	2.0%	✓
ObjaverseV1[10]	Object	167K	Synthetic	4.6%	

Table A.2: Summary of labeled training frame counts and pretrained backbones for the models compared in this paper.

Method	#Total Training Frames	Pretrained Backbone
ZoeDepth [5]	~ 2M	MiDaS BEiT384-L [47]
DA V1 [67]	1.5M (+ 62M pseudo-labeled)	DINOv2 ViT-Large
DA V2 [68]	595K (+ 62M pseudo-labeled)	DINOv2 ViT-Large
Metric3D V2 [23]	16M	DINOv2 ViT-Large
UniDepth V1 [44]	3.7M	DINOv2 ViT-Large
UniDepth V2 [45]	16M	DINOv2 ViT-Large
Depth Pro [7]	~ 6M	DINOv2 ViT-Large
MoGe [61]	9M	DINOv2 ViT-Large
<i>Ours</i>	8.9M	DINOv2 ViT-Large

A.3 Evaluation Protocol

Relative Geometry We follow the evaluation protocol of alignment in MoGe [61]. Predictions and ground truth are aligned in scale (and shift, if applicable) for each image before measuring errors as specified below

- **Scale-invariant point map.** The scale a^* to align prediction with ground truth is computed as:

$$a^* = \operatorname{argmin}_a \sum_{i \in \mathcal{M}} \frac{1}{z_i} \|a\hat{\mathbf{p}}_i - \mathbf{p}_i\|_1, \quad (9)$$

- **Affine-invariant point map.** The scale a^* and shift $\mathbf{b}^*$ are computed as:

$$(a^*, \mathbf{b}^*) = \operatorname{argmin}_{a, \mathbf{b}} \sum_{i \in \mathcal{M}} \frac{1}{z_i} \|a\hat{\mathbf{p}}_i + \mathbf{b} - \mathbf{p}_i\|_1. \quad (10)$$

- **Scale-invariant depth map**, the scale a^* is computed as

$$a^* = \operatorname{argmin}_s \sum_{i \in \mathcal{M}} \frac{1}{z_i} |a\hat{z}_i - z_i|. \quad (11)$$

- **Affine-invariant depth map**. The scale a^* and shift b^* are computed as

$$(a^*, b^*) = \operatorname{argmin}_s \sum_{i \in \mathcal{M}} \frac{1}{z_i} |a\hat{z}_i + b - z_i|. \quad (12)$$

- **Affine-invariant disparity map**. We follow the established protocol for affine disparity alignment [47], using least-squares to align predictions in disparity space:

$$(a^*, b^*) = \operatorname{argmin}_s \sum_{i \in \mathcal{M}} (a\hat{d}_i + b - d_i)^2, \quad (13)$$

where $\hat{d}_i$ is the predicted disparity and d_i is the ground truth, defined as $d_i = 1/z_i$. To prevent aligned disparities from taking excessively small or negative values, the aligned disparity is truncated by the inverted maximum depth $1/z_{\max}$ before inversion. The final aligned depth $\hat{z}_i^*$ is computed as:

$$\hat{z}_i^* := \frac{1}{\max(a^*\hat{d}_i + b^*, 1/z_{\max})}. \quad (14)$$

Metric Geometry

- **Metric depth**. The output is evaluated without alignment and clamping range of values for all methods, unless specific post-processing is hard-coded in its model inference pipeline.
- **Metric point map**. The point map prediction is aligned with the ground truth by the optimal translation:

$$b^* = \operatorname{argmin}_b \sum_{i \in \mathcal{M}} \frac{1}{z_i} \|\hat{p}_i + b - p_i\|_1. \quad (15)$$

B Additional Experiments and Results

B.1 Test-time Resolution Scaling

In ViT-based models, the native input resolution is determined by the number of image tokens derived from fixed-size patches, specifically, 14^2 for DINOv2 models. As such, resolution scaling can be effectively studied through varying token counts. Our model is trained across a wide range of token counts from 1200 to 3600, corresponding to native input resolutions ranging approximately from 484^2 to 1188^2 . This training setup enables robust generalization to a broad range of resolutions and flexible usage with details as follows.

Geometry Accuracy MoGe [61] and UniDepth V2 [44] are both trained on diverse input resolutions and aspect ratios, which helps them maintain accuracy under resolution shifts within a moderate range (1200 - 3000). In contrast, models such as Depth Anything [67, 68] and Metric3D V2 [23] are trained with fixed input resolution and exhibit substantial performance degradation when evaluated at resolutions that diverge from their training setting. Our method, trained over a broader resolution spectrum, remains robust under test-time scaling. As shown in Fig. B.3a, it maintains the top accuracy when scaled up for improved detail or down for faster inference—even beyond the training range.

Boundary Sharpness Higher input resolutions and more image tokens generally lead to sharper boundaries in dense prediction tasks, as observed in prior works [68, 48, 29] and also shown in Fig. B.2. In Fig. B.3b, we evaluate several DINOv2-based methods for boundary sharpness at different test-time resolutions. Note that Depth Pro operates at a fixed high resolution due to its specialized multi-scale, patch-based architecture. Our approach consistently delivers the sharpest predictions at each resolution and outperforms Depth Pro using significantly fewer tokens to reach similar levels of detail.

Latency Trade-off As shown in Fig. B.3c, inference latency scales roughly linearly with the number of tokens. Although all compared methods share the same ViT backbone, overall runtime can vary due to differences in decoder complexity and architectural choices. Our model adopts a lightweight design that enables fast inference while maintaining strong accuracy, achieving a favorable trade-off between latency and performance across a wide range of resolutions—within a single unified framework.

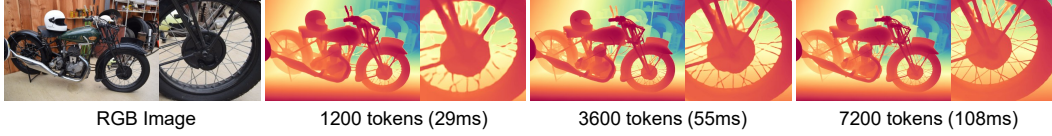


Figure B.2: Trading latency for improved visual sharpness by increasing image tokens.

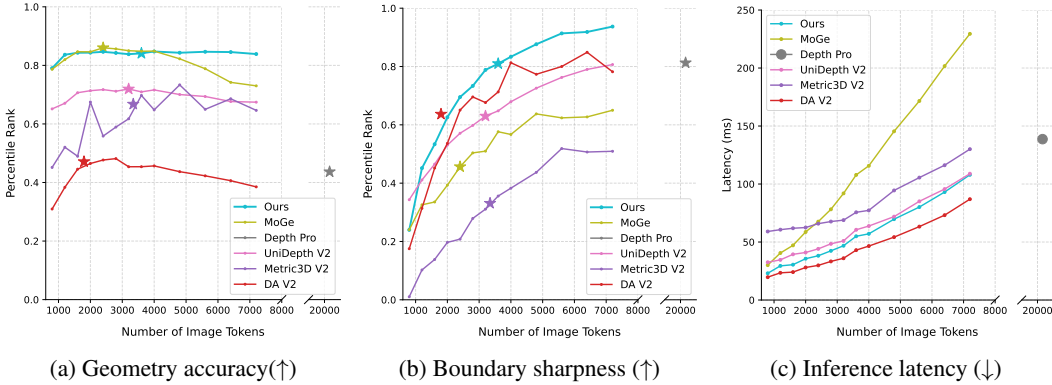


Figure B.3: Performance comparison under test-time resolution scaling. ★ denotes the default configuration for each method. (a) Percentile rank ($\frac{x - \text{worst}}{\text{best} - \text{worst}}$) averaged across all evaluated datasets and two geometry metrics (metric and relative geometry accuracy). (b) Average percentile rank for boundary sharpness. Both are evaluated on a 1/10 subset uniformly sampled from the evaluation benchmarks. (c) Inference latency measured on an NVIDIA A100 GPU with FP16 precision. Our method demonstrates the most favorable balance between latency and performance across different resolutions.

B.2 Runtime Analysis

As shown in Table B.3, we evaluate the runtime performance of each method under their representative test-time configurations. Specifically, we measure single-frame inference latency and peak GPU memory usage on an NVIDIA A100 GPU. These metrics provide a practical comparison of computational efficiency and resource requirements across different architectures.

B.3 More Visual Results

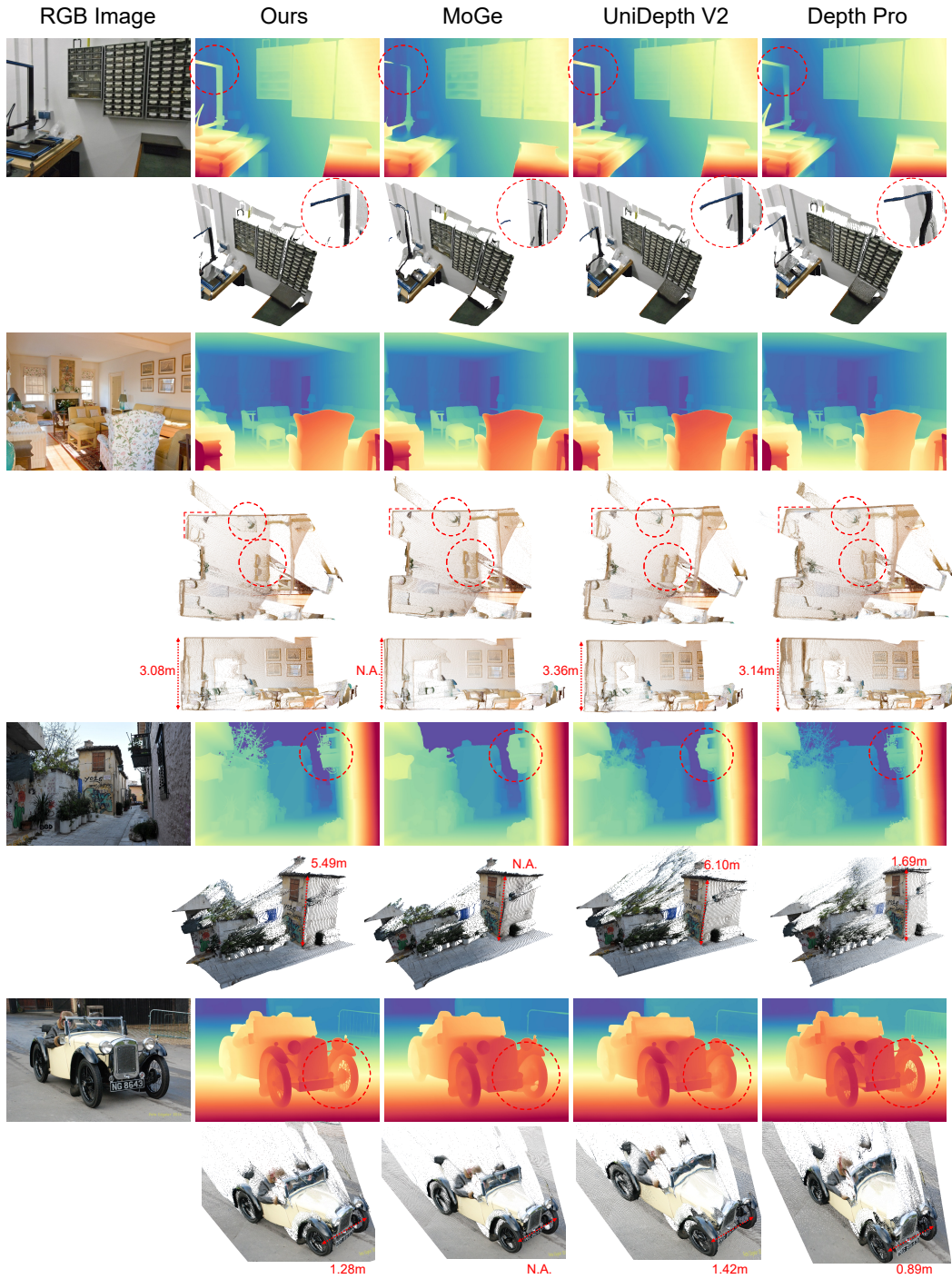
More visual results for qualitative comparison are included in Fig. B.4 and Fig. B.5. Representative failure cases are shown in Fig. B.6.

B.4 Complete Evaluation on Individual Datasets

In the paper, we only listed the average performance across multiple datasets for qualitative comparison and ablation study. Table B.4 and Table B.5 list all the results for each individual datasets.

Table B.3: Runtime statistics measured on a single NVIDIA A100 GPU for single-frame inference.

Method	#Parameters	#Tokens	Native Resolution	Latency (ms)		Memory (GB)	
				FP16	FP32	FP16	FP32
DA V2	335M	1369	518 ²	24	86	0.91	1.8
Metric3D V2	412M	3344	1064×616	87	255	1.4	2.3
UniDepth V2	354M	1020	448 ²	33	84	1.1	1.8
		3061	774 ²	50	206	1.8	2.5
Depth Pro	504M	20160	1536 ²	139	906	3.7	8.0
MoGe	314M	1200	484 ²	40	93	0.74	1.4
		2500	700 ²	70	192	0.88	1.6
<i>Ours</i>	326M	1200	484 ²	29	82	0.96	1.7
		2500	700 ²	39	157	1.1	2.1
		3600	840 ²	55	238	1.3	2.5
		7200	1188 ²	108	565	1.9	3.8



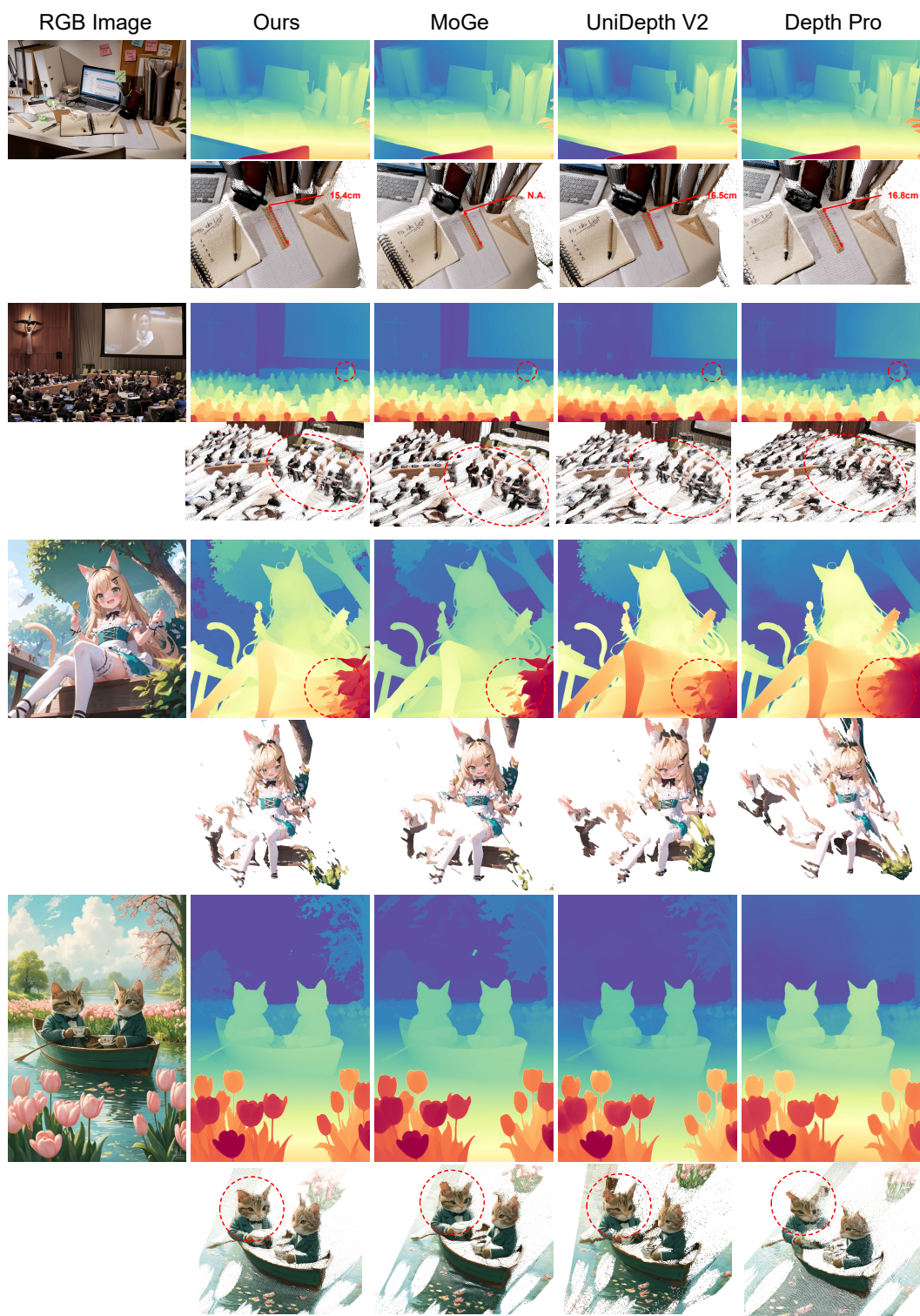


Figure B.5: More visual results on open-domain images (2/2). *Best viewed zoomed in.*

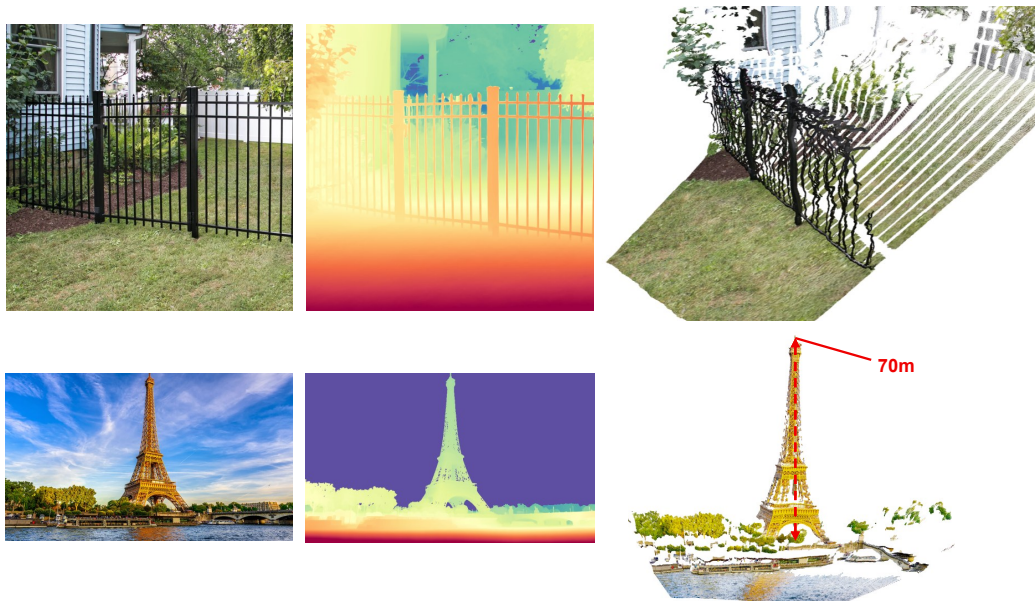


Figure B.6: Failure cases. *Top*: the reconstruction of very thin structures appears distorted, even though the predicted depth maps look sharp. *Bottom*: the model predicts a height of 70 m for the Eiffel Tower, far below its true 330 m. Such scale errors occur when the scene contains atypical content or lacks familiar geometric cues.

Method	NYUv2 Rel↓ δ_1 ↑	KITTI Rel↓ δ_1 ↑	ETH3D Rel↓ δ_1 ↑	iBims-1 Rel↓ δ_1 ↑	GSO Rel↓ δ_1 ↑	Sintel Rel↓ δ_1 ↑	DDAD Rel↓ δ_1 ↑	DIODE Rel↓ δ_1 ↑	Spring Rel↓ δ_1 ↑	HAMMER Rel↓ δ_1 ↑	Avg. Rel↓ δ_1 ↑ Rank↓
Metric point map											
MASt3R	7.11 95.6	26.0 45.8	27.4 43.1	10.1 89.3	- -	- -	35.4 28.7	21.7 66.3	- -	56.0 18.3	26.2 55.3 4.93
UniDepth V1	4.80 98.3	4.52 98.5	22.4 63.1	10.8 92.8	- -	- -	11.4 89.5	12.8 88.9	- -	18.0 79.5	12.1 87.2 2.71
UniDepth V2	4.83 98.0	5.88 97.5	9.46 95.0	5.23 97.9	- -	- -	13.3 90.3	17.0 80.8	- -	15.0 83.9	10.1 91.9 2.43
Depth Pro	6.13 97.3	11.1 85.3	21.2 64.9	6.89 96.9	- -	- -	22.6 61.3	13.5 81.8	- -	14.5 86.0	13.7 81.9 3.29
Ours	4.44 98.3	7.44 94.4	7.19 97.7	5.63 97.4	- -	- -	11.4 87.9	7.85 92.3	- -	13.4 87.0	8.19 93.6 1.64
Metric depth map (wo/ GT intrinsics)											
ZoeDepth	11.0 91.9	17.0 85.4	57.1 33.7	17.4 67.2	- -	- -	38.9 38.6	39.3 29.3	- -	94.3 3.23	39.3 49.9 5.90
MASt3R	10.8 92.7	56.7 9.84	47.2 20.1	18.7 61.5	- -	- -	62.4 5.51	54.9 19.0	- -	97.2 6.74	49.7 30.3 6.71
DA V1	10.5 94.9	11.6 94.5	40.2 24.0	12.9 81.8	- -	- -	34.5 44.7	58.0 16.2	- -	54.8 27.3	31.8 54.8 5.50
DA V2	16.4 80.9	10.6 88.6	36.1 36.3	11.1 91.7	- -	- -	41.7 37.5	41.2 22.1	- -	52.1 38.9	29.9 56.6 4.43
UniDepth V1	7.59 97.6	4.69 98.4	56.9 14.9	23.8 57.6	- -	- -	13.8 85.1	17.1 71.9	- -	38.2 46.7	23.2 67.5 3.32
UniDepth V2	10.6 92.8	8.58 95.4	20.7 69.5	9.52 93.2	- -	- -	18.4 77.6	43.0 51.8	- -	38.2 46.8	21.3 75.3 2.54
Depth Pro	10.7 91.9	23.5 38.3	38.5 32.8	15.9 81.5	- -	- -	33.4 35.3	31.9 37.7	- -	39.1 63.0	27.6 54.4 4.36
Ours	7.33 96.1	18.1 62.9	10.4 90.8	13.6 83.0	- -	- -	15.8 73.0	17.5 66.4	- -	26.9 65.6	15.7 76.8 2.21
Metric depth map (w/ GT intrinsics)											
Metric3D V2	7.16 96.5	5.25 98.0	11.8 88.8	9.96 94.1	- -	- -	9.21 93.7	49.1 1.98	- -	35.7 44.3	18.3 73.9 2.75
UniDepth V1	5.98 97.9	4.43 98.5	44.5 26.7	22.6 60.5	- -	- -	13.0 87.2	21.0 63.5	- -	38.6 45.9	21.4 68.6 2.50
UniDepth V2	7.81 96.0	5.98 97.7	15.0 85.2	7.71 95.5	- -	- -	14.1 89.3	41.0 67.1	- -	37.7 47.1	18.5 82.6 2.57
Ours	6.46 96.9	8.64 93.7	10.5 92.2	9.92 92.4	- -	- -	13.1 85.6	16.2 77.1	- -	30.4 74.2	13.6 87.4 2.00
Scale-invariant point map											
MASt3R	6.26 96.0	10.0 93.8	6.28 95.5	7.55 95.1	5.03 99.0	31.5 50.2	15.9 77.6	12.8 85.0	39.3 33.7	10.7 95.0	14.5 82.1 5.45
UniDepth V1	5.33 98.4	5.96 98.5	18.5 77.6	5.29 97.4	6.58 99.6	33.0 48.9	11.4 90.2	12.3 91.0	33.1 49.8	4.83 98.5	13.6 85.0 3.83
UniDepth V2	5.59 98.3	5.41 98.0	6.58 97.2	5.56 98.1	4.53 99.7	27.2 56.3	13.4 91.2	12.0 93.4	31.9 46.0	4.20 99.2	11.6 87.7 2.98
Depth Pro	5.04 97.7	10.6 95.1	11.2 92.0	5.84 97.1	4.94 99.8	26.9 63.9	15.8 81.0	8.52 91.6	28.1 60.5	6.82 98.7	12.4 87.7 3.83
MoGe	4.86 98.4	5.47 97.4	4.58 98.9	4.63 97.1	2.58 100	22.3 69.5	12.3 90.3	6.58 94.5	4.84 96.4	6.45 98.1	7.46 94.1 2.14
Ours	3.94 98.3	8.27 97.5	5.45 98.6	5.34 98.3	2.55 100	23.1 66.8	11.0 90.7	8.42 93.7	31.1 42.4	8.77 98.4	10.8 88.5 2.40
Affine-invariant point map											
MASt3R	5.30 96.3	8.32 92.3	5.48 96.6	5.72 95.0	3.50 99.2	26.3 62.8	14.7 79.6	8.10 90.1	33.3 51.1	5.34 96.6	11.6 86.0 5.45
UniDepth V1	3.93 98.4	4.29 98.6	12.2 89.6	4.65 98.0	2.99 99.8	28.5 58.4	10.3 90.5	8.56 90.9	29.6 58.5	4.15 98.7	10.9 88.1 3.95
UniDepth V2	3.66 98.4	4.75 98.0	4.35 98.4	4.05 98.1	2.91 99.9	17.9 76.5	12.0 90.8	7.45 92.4	25.1 66.9	3.45 99.4	8.56 91.9 2.55
Depth Pro	4.36 97.9	9.15 90.7	7.73 94.0	4.34 97.4	3.16 99.7	19.6 74.5	14.4 81.2	6.28 93.7	25.0 66.0	5.31 98.8	9.93 89.4 4.30
MoGe	3.68 98.3	4.86 97.2	3.57 99.0	3.61 97.3	1.14 100	16.8 77.8	10.5 91.4	4.37 96.4	4.45 96.4	3.88 98.1	5.69 95.2 2.14
Ours	3.33 98.4	6.47 96.4	3.89 98.7	3.65 98.5	1.16 100	17.4 77.0	10.1 90.3	5.13 94.9	24.5 63.7	4.19 99.1	7.98 91.7 2.23
Local point map											
MASt3R	- -	- -	5.54 95.3	6.19 95.0	- -	- -	11.4 87.9	8.58 91.8	8.75 90.9	- -	8.09 92.2 5.40
UniDepth V1	- -	- -	8.61 92.6	5.92 96.0	- -	- -	13.4 84.3	8.18 92.0	9.95 90.0	- -	9.21 91.0 5.55
UniDepth V2	- -	- -	3.99 97.4	4.02 97.3	- -	- -	9.35 92.2	8.18 92.4	6.15 95.3	- -	6.34 94.9 3.10
Depth Pro	- -	- -	4.76 96.9	4.11 97.5	- -	- -	10.8 89.5	8.08 92.4	6.80 94.1	- -	6.91 94.1 3.55
MoGe	- -	- -	3.21 98.1	4.16 96.8	- -	- -	8.63 92.7	6.74 94.3	4.78 96.3	- -	5.50 95.6 2.05
Ours	- -	- -	3.27 98.2	3.61 97.7	- -	- -	8.13 93.2	6.57 94.3	5.09 96.1	- -	5.33 95.9 1.35
Scale-invariant depth map											
ZoeDepth	5.62 96.3	7.27 91.9	10.4 87.3	7.45 93.2	3.23 99.9	27.4 61.8	17.0 72.8	11.3 85.2	30.3 55.9	7.42 94.7	12.7 83.9 8.75
MASt3R	5.37 96.0	6.24 94.5	5.68 95.5	5.58 95.2	3.72 99.1	26.3 63.7	13.5 81.5	8.37 89.4	32.2 53.5	5.50 96.5	11.2 86.5 7.65
DA V1	4.77 97.5	5.61 95.6	9.41 88.9	5.53 95.8	5.49 99.3	28.3 56.7	13.2 81.5	10.3 87.5	27.3 59.1	6.88 96.4	11.7 85.8 8.22
DA V2	5.03 97.3	7.23 93.7	6.12 95.5	4.32 97.9	4.38 99.3	23.0 65.2	14.7 78.0	7.95 90.0	28.0 61.1	5.92 97.7	10.7 87.6 6.80
Metric3D V2	4.69 97.4	4.00 98.5	3.84 98.5	4.23 97.7	2.46 99.9	20.7 69.8	7.41 94.6	3.29 98.4	24.4 64.4	4.19 99.1	7.92 91.8 3.39
UniDepth V1	3.86 98.4	3.73 98.6	5.67 97.0	4.79 97.4	4.18 99.7	28.3 58.8	10.1 90.5	6.83 92.8	29.2 59.3	4.19 98.4	10.1 89.1 5.12
UniDepth V2	3.65 98.4	4.24 98.0	3.23 98.9	3.45 98.1	3.16 99.7	23.1 65.3	11.0 91.5	5.92 94.1	24.9 65.1	3.48 99.1	8.61 90.8 3.10
Depth Pro	4.42 97.6	5.47 96.2	7.54 94.1	4.13 97.4	2.18 99.9	23.9 68.7	14.0 82.0	7.05 92.0	25.1 63.8	4.36 98.9	9.81 89.1 5.33
MoGe	3.44 98.4	4.25 97.8	3.36 98.9	3.46 97.0	1.47 100	19.3 73.4	9.17 90.5	4.89 94.7	4.63 96.4	3.77 98.1	5.77 94.5 2.72
Ours	3.44 98.2	4.11 98.0	3.55 98.7	3.16 98.2	1.49 100	19.6 71.6	8.91 91.2	5.30 94.6	20.0 72.4	3.96 99.2	7.35 92.2 2.12
Affine-invariant depth											
ZoeDepth	4.76 97.3	5.59 95.1	7.27 94.2	5.85 95.7	2.54 99.9	21.8 69.2	14.2 80.1	7.80 90.9	24.3 66.6	6.65 95.7	10.1 88.5 9.09
MASt3R	4.67 96.7	5.79 95.1	4.64 97.0	4.62 95.6	2.85 99.4	21.3 70.3	12.5 83.4	5.79 94.1	27.4 62.8	4.21 96.8	9.38 89.1 7.97
DA V1	3.82 98.3	5.04 96.4	6.23 95.2	4.23 97.3	1.98 100	20.1 71.8	11.3 86.1	6.75 92.6	22.4 68.9	5.77 97.3	8.76 90.4 6.91
DA V2	4.16 97.9	6.77 94.3	4.63 97.2	3.44 98.3	1.44 100	17.1 76.6	13.4 81.8	5.41 94.6	23.7 68.7	4.73 98.9	8.48 90.8 6.15
Metric3D V2	3.94 97.6	3.50 98.4	3.24 99.0	3.28 98.3	2.10 99.4	26.6 71.7	7.15 94.8	2.75 98.7	21.0 72.5	3.02 99.0	7.66 92.9 4.53
UniDepth V1	3.40 98.6	3.55 98.7	4.92 97.5	3.76 98.2	2.48 99.9	24.9 64.1	9.46 90.8	4.90 96.2	25.2 67.3	3.55 98.9	8.61 91.0 5.67
UniDepth V2	2.96 98.6	3.85 98.1	2.95 98.5	2.64 98.4	1.37 100	13.3 83.2	10.5 90.9	4.05 96.5	20.1 75.4	2.48 99.6	6.42 93.9 2.80
Depth Pro	3.67 98.2	5.12 96.8	4.97 96.4	3.23 98.3	1.46 100	15.8 80.1	12.6 84.1	4.66 95.6	21.7 70.5	3.30 99.6	7.65 92.0 5.05
MoGe	2.92 98.6	3.94 98.0	2.69 99.2	2.74 97.9	0.94 100	13.0 83.2	8.40 92.1	3.16 97.5	4.34 96.4	3.00 98.3	4.51 96.1 2.94
Ours	2.89 98.6	3.75 98.1	2.80 99.1	2.36 98.8	0.94 100	13.3 82.5	8.26 92.5	3.14 97.4	15.9 81.2	2.85 99.3	5.62 94.8 2.02
Affine-invariant disparity											
ZoeDepth	5.21 97.7	5.84 95.6	8.07 94.0	6.19 96.1	2.60 99.9	26.9 66.3	14.1 81.7	8.17 92.0	27.2 63.0	6.84 96.4	11.1 88.3 8.78
DA V1	4.20 98.4	5.40 97.0	4.68 98.2	4.18 97.6	1.54 100	20.2 77.6	12.7 86.9	5.69 95.7	22.2 72.5	5.56 98.0	8.63 92.2 5.62
DA V2	4.14 98.3	5.61 96.7	4.71 97.9	3.47 98.5	1.24 100	21.4 72.8	13.1 86.4	5.29 96.1	24.3 70.6	4.97 99.1	8.82 91.6 5.42
Metric3D V2	13.4 81.5	3.76 98.2	4.30 97.7	8.55 92.3	1.80 100	21.8 72.4	7.35 94.1	7.70 90.2	23.3 68.1	3.17 99.2	9.51 89.4 6.17
MASt3R	5.07 96.8	5.93 95.5	5.25 96.4	5.39 95.7	2.98 99.7	30.2 65.1	13.0 83.6	6.41 94.3	37.3 53.2	4.41 97.2	11.6 87.8 8.60
UniDepth V1	3.78 98.7	3.64 98.7	5.34 97.2	4.06 98.1	2.56 99.9	28.6 60.7	9.94 89.1	5.95 95.5	30.0 61.6	3.64 99.1	9.75 89.9 5.92
UniDepth V2	3.38 98.7	3.99 98.0	2.97 99.0	3.15 98.3	1.30 100	17.2 79.9	10.2 90.2	4.43 96.4	24.4 69.6	2.51 99.6	7.35 93.0 2.75
Depth Pro	4.21 98.1	5.10 97.0	4.94 96.7	3.74 98.2	1.49 100	17.4 79.1	11.7 87.1	4.84 96.4	27.5 64.5	3.31 99.6	8.42 91.7 5.08
MoGe	3.38 98.6	4.05 98.1	3.11 98.9	3.23 98.0	0.96 100	18.4 79.5	8.99 91.5	3.98 97.2	6.43 93.7	3.30 98.8	5.58 95.4 3.17
Ours	3.35 98.6	3.92 98.1	3.21 98.9	2.85 98.7	0.96 100	18.0 78.7	8.69 92.1	4.03 97.2	18.7 76.6	2.90 99.5	6.66 93.8 2.17

Table B.4: Evaluation results of baselines and our method on each dataset.

Ablation		NYUv2	KITTI	ETH3D	iBims-1	GSO	Sintel	DDAD	DIODE	Spring	HAMMER	Avg.
Data	Scale Prediction	Rel↓ δ_1 ↑	Rel↓ δ_1 ↑	Rel↓ δ_1 ↑	Rel↓ δ_1 ↑	Rel↓ δ_1 ↑	Rel↓ δ_1 ↑	Rel↓ δ_1 ↑	Rel↓ δ_1 ↑	Rel↓ δ_1 ↑	Rel↓ δ_1 ↑	Rel↓ δ_1 ↑
Metric point map												
Improved real	Entangled (SI-Log)	6.00 97.3	8.33 93.4	11.6 89.4	7.78 94.6	- -	- -	14.4 83.1	10.6 88.4	- -	11.4 88.7	10.0 90.7
Improved real	Entangled (Shift inv.)	5.26 97.6	8.81 92.6	10.4 91.7	6.14 96.6	- -	- -	13.0 84.8	8.97 91.0	- -	10.4 90.3	9.00 92.1
Improved real	Decoupled (Conv)	5.37 97.8	9.56 91.8	9.46 94.1	6.49 95.6	- -	- -	13.3 83.5	8.93 91.9	- -	14.2 85.1	9.62 91.4
Synthetic only	Decoupled (MLP)	8.58 94.7	9.48 91.9	14.9 83.4	8.20 94.2	- -	- -	16.5 80.4	11.0 88.8	- -	18.4 78.2	12.4 87.4
Raw real	Decoupled (MLP)	5.36 97.8	7.70 94.5	8.58 94.6	6.60 95.7	- -	- -	12.2 85.5	9.01 91.5	- -	13.7 85.4	9.02 92.1
Improved real	Decoupled (MLP)	5.47 97.6	8.98 92.6	8.75 94.3	6.24 96.1	- -	- -	12.8 84.6	9.26 90.9	- -	12.9 87.4	9.20 91.9
Metric depth map (w/ GT intrinsics)												
Improved real	Entangled (SI-Log)	9.65 91.4	14.5 77.3	16.4 73.7	20.1 56.2	- -	- -	19.1 67.7	22.3 54.3	- -	23.0 59.8	17.9 68.6
Improved real	Entangled (Shift inv.)	9.04 93.1	19.1 56.8	15.5 76.8	15.1 72.1	- -	- -	18.0 67.9	19.9 59.8	- -	22.0 55.3	16.9 68.8
Improved real	Decoupled (Conv)	9.22 92.7	20.3 51.8	13.8 79.8	15.8 71.0	- -	- -	18.1 66.6	19.0 61.8	- -	27.5 54.9	17.7 68.4
Synthetic only	Decoupled (MLP)	18.1 73.7	15.8 71.0	24.7 53.2	15.8 76.6	- -	- -	21.9 62.4	22.7 56.5	- -	32.8 62.0	21.7 65.1
Raw real	Decoupled (MLP)	9.22 92.9	13.8 80.5	13.8 82.1	16.7 72.4	- -	- -	16.5 72.3	19.7 61.5	- -	20.8 67.9	15.8 75.7
Improved real	Decoupled (MLP)	9.48 92.2	18.7 59.2	13.5 82.6	13.6 79.3	- -	- -	17.0 69.2	20.0 59.7	- -	23.4 67.4	16.5 72.8
Scale-invariant point map												
Improved real	Entangled (SI-Log)	6.03 97.4	9.68 95.3	8.13 95.3	8.63 96.8	4.01 100	26.6 59.4	13.8 84.8	10.3 89.8	31.2 48.0	10.4 95.6	12.9 86.2
Improved real	Entangled (Shift inv.)	5.00 97.8	10.7 95.8	7.02 96.7	7.42 97.4	3.42 100	26.0 58.2	12.7 86.6	8.97 92.2	28.9 49.8	10.5 97.5	12.1 87.2
Improved real	Decoupled (Conv)	4.84 97.8	12.1 94.8	6.55 96.9	7.15 96.9	3.19 100	26.3 55.8	12.9 85.5	9.11 91.8	29.9 46.9	10.4 97.0	12.2 86.3
Synthetic only	Decoupled (MLP)	6.66 96.9	11.3 93.0	6.85 95.8	5.99 96.7	3.14 100	25.4 61.3	15.0 81.1	10.5 89.4	30.9 46.8	7.39 97.5	12.3 85.8
Raw real	Decoupled (MLP)	4.88 98.0	9.15 96.0	6.08 97.1	7.31 96.8	3.06 100	24.8 60.8	11.8 87.7	8.34 92.3	28.1 53.2	10.9 96.2	11.4 87.8
Improved real	Decoupled (MLP)	5.00 97.8	11.2 95.0	6.21 97.4	6.52 97.3	2.97 100	25.6 60.3	12.6 87.0	8.76 92.3	28.3 51.0	9.10 98.3	11.6 87.6
Affine-invariant point map												
Improved real	Entangled (SI-Log)	5.00 97.7	8.22 93.0	6.72 96.1	5.71 96.8	2.57 100	21.1 71.0	12.9 84.5	7.36 91.9	26.7 59.7	6.37 97.1	10.3 88.8
Improved real	Entangled (Shift inv.)	4.17 97.9	8.58 93.0	5.42 97.0	4.74 96.8	1.78 100	19.7 72.7	11.7 86.6	6.07 93.5	23.3 66.3	5.02 98.6	9.05 90.2
Improved real	Decoupled (Conv)	4.08 98.0	9.65 90.8	5.12 97.0	4.67 97.0	1.66 100	19.6 72.2	12.0 85.4	6.11 93.4	23.4 67.9	5.24 98.0	9.15 90.0
Synthetic only	Decoupled (MLP)	5.48 97.0	9.11 90.2	5.93 96.2	4.94 96.6	1.56 100	20.0 73.1	13.7 81.9	7.01 91.6	25.5 63.7	4.44 98.7	9.77 88.9
Raw real	Decoupled (MLP)	4.06 98.2	7.37 94.5	4.89 97.5	4.74 96.8	1.61 100	19.0 73.6	10.9 87.9	5.89 93.7	23.4 67.0	5.09 98.2	8.70 90.7
Improved real	Decoupled (MLP)	4.14 98.0	8.95 92.0	4.94 97.5	4.50 97.2	1.62 100	19.6 73.6	11.7 86.6	6.06 93.3	22.8 68.7	4.40 98.9	8.87 90.6
Local point map												
Improved real	Entangled (SI-Log)	- -	- -	6.30 95.6	5.96 96.6	- -	12.0 87.5	8.14 92.5	8.63 92.5	- -	- -	8.21 92.9
Improved real	Entangled (Shift inv.)	- -	- -	4.61 97.2	4.56 97.2	- -	10.3 90.2	7.38 93.5	6.61 94.7	- -	- -	6.69 94.6
Improved real	Decoupled (Conv)	- -	- -	4.25 97.5	4.34 97.3	- -	9.72 91.0	7.24 93.6	6.17 95.1	- -	- -	6.34 94.9
Synthetic only	Decoupled (MLP)	- -	- -	4.37 97.4	4.45 97.2	- -	9.33 91.7	7.51 93.3	6.44 94.9	- -	- -	6.42 94.9
Raw real	Decoupled (MLP)	- -	- -	4.28 97.4	4.55 97.1	- -	9.64 91.2	7.11 93.7	6.28 94.9	- -	- -	6.37 94.9
Improved real	Decoupled (MLP)	- -	- -	4.20 97.5	4.31 97.3	- -	9.34 91.9	7.21 93.7	6.21 95.0	- -	- -	6.25 95.1
Scale-invariant depth map												
Improved real	Entangled (SI-Log)	4.99 97.4	5.18 96.7	6.48 95.2	5.26 97.3	2.64 100	23.2 65.7	11.6 86.2	7.76 91.3	24.8 63.4	6.40 96.7	9.83 89.0
Improved real	Entangled (Shift inv.)	4.17 97.8	4.57 97.5	5.03 96.7	4.42 97.6	2.09 100	22.4 66.7	10.3 88.0	6.16 93.5	20.8 69.2	4.69 98.6	8.46 90.6
Improved real	Decoupled (Conv)	4.08 97.9	4.61 97.3	4.80 96.9	4.32 97.1	1.92 100	22.5 64.9	10.3 87.7	6.26 93.2	21.0 68.8	4.81 98.2	8.46 90.2
Synthetic only	Decoupled (MLP)	5.05 96.9	5.47 96.3	5.64 95.9	4.76 96.9	1.90 100	21.7 68.0	12.0 85.1	7.16 91.2	22.2 67.4	4.66 97.9	9.05 89.6
Raw real	Decoupled (MLP)	4.09 98.0	4.60 97.2	4.82 97.1	4.49 97.0	1.92 100	21.8 66.5	9.79 88.4	6.09 93.2	21.8 68.6	4.67 98.1	8.41 90.4
Improved real	Decoupled (MLP)	4.16 97.8	4.59 97.2	4.62 97.4	4.20 97.4	1.89 100	21.9 67.8	10.1 88.4	6.08 93.3	20.4 71.6	4.34 98.9	8.23 91.0
Affine-invariant depth												
Improved real	Entangled (SI-Log)	4.32 98.0	4.91 97.0	5.21 96.9	4.15 97.8	2.11 100	17.4 76.3	10.8 87.9	5.17 95.7	20.3 72.6	5.26 97.6	7.96 92.0
Improved real	Entangled (Shift inv.)	3.57 98.2	4.26 97.6	3.98 97.9	3.35 98.2	1.44 100	16.0 78.4	9.58 89.7	4.01 96.8	17.7 76.7	3.58 98.7	6.75 93.2
Improved real	Decoupled (Conv)	3.48 98.3	4.31 97.4	3.77 98.1	3.23 98.0	1.35 100	15.2 79.2	9.48 89.5	3.99 96.4	17.7 76.4	3.74 98.5	6.62 93.2
Synthetic only	Decoupled (MLP)	4.19 97.8	5.04 96.6	4.40 97.6	3.73 97.7	1.27 100	15.6 78.9	11.1 86.8	4.50 95.7	19.1 75.0	3.58 98.8	7.25 92.5
Raw real	Decoupled (MLP)	3.48 98.4	4.29 97.3	3.75 98.3	3.37 98.0	1.32 100	15.2 80.1	8.95 90.2	3.91 96.7	18.6 75.7	3.48 98.4	6.63 93.3
Improved real	Decoupled (MLP)	3.53 98.3	4.30 97.3	3.69 98.3	3.16 98.3	1.33 100	15.3 79.3	9.28 90.1	3.95 96.6	17.6 76.8	3.19 99.0	6.53 93.4
Affine-invariant disparity												
Improved real	Entangled (SI-Log)	4.69 98.1	4.99 97.2	5.73 96.7	4.63 97.8	2.14 100	21.9 71.1	11.3 88.0	5.69 95.8	23.8 68.4	5.46 98.3	9.03 91.1
Improved real	Entangled (Shift inv.)	4.03 98.3	4.39 97.7	4.48 97.8	3.85 98.2	1.47 100	20.2 73.8	9.84 89.8	4.75 96.5	21.3 69.5	3.74 99.0	7.80 92.1
Improved real	Decoupled (Conv)	3.96 98.4	4.49 97.5	4.29 97.9	3.75 98.4	1.39 100	19.9 74.4	9.89 89.7	4.75 96.3	21.1 70.0	3.81 98.8	7.73 92.1
Synthetic only	Decoupled (MLP)	4.75 97.8	5.17 96.7	4.86 97.6	4.29 98.0	1.31 100	20.8 74.0	11.5 87.3	5.21 95.9	22.2 69.7	3.66 99.0	8.38 91.6
Raw real	Decoupled (MLP)	3.92 98.4	4.50 97.4	4.23 98.0	3.95 98.2	1.35 100	19.9 74.8	9.22 90.6	4.78 96.3	21.4 69.6	3.61 98.7	7.69 92.2
Improved real	Decoupled (MLP)	4.03 98.3	4.45 97.5	4.11 98.1	3.74 98.2	1.37 100	19.8 75.4	9.71 90.1	4.79 96.2	20.0 72.5	3.30 99.3	7.53 92.6

Table B.5: Evaluation results of ablation study on each sets