

# InfiniDepth: Arbitrary-Resolution and Fine-Grained Depth Estimation with Neural Implicit Fields

Hao Yu<sup>1,2\*</sup> Haotong Lin<sup>1\*</sup> Jiawei Wang<sup>1\*</sup> Jiaxin Li<sup>1</sup> Yida Wang<sup>2</sup> Xueyang Zhang<sup>2</sup>  
 Yue Wang<sup>1</sup> Xiaowei Zhou<sup>1</sup> Ruizhen Hu<sup>3</sup> Sida Peng<sup>1†</sup>

<sup>1</sup>Zhejiang University <sup>2</sup>Li Auto <sup>3</sup>Shenzhen University

Project Page: [zju3dv.github.io/InfiniDepth](https://zju3dv.github.io/InfiniDepth)

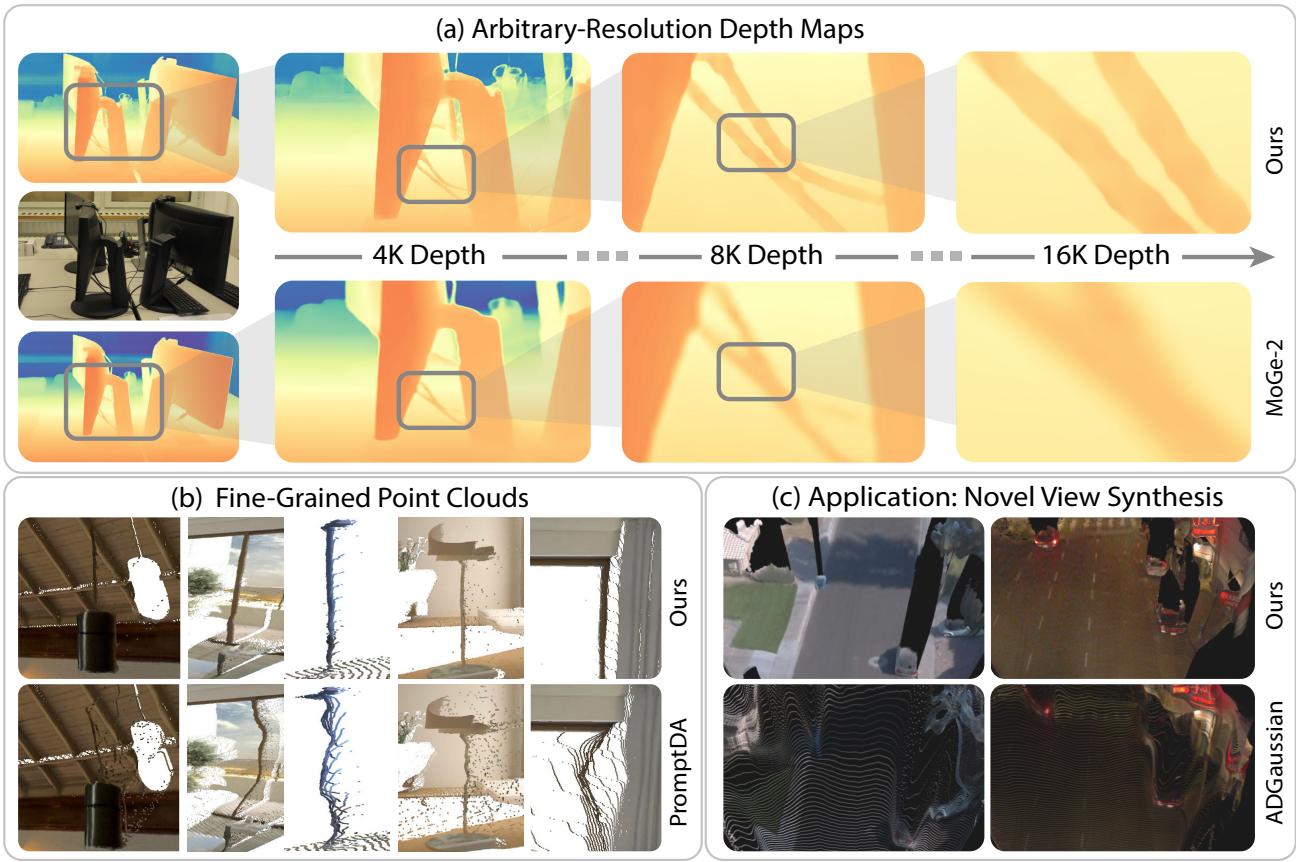


Figure 1. **InfiniDepth** is a new depth representation which models depth as neural implicit fields, enabling arbitrary-resolution and fine-grained depth estimation. It also benefits novel view synthesis under large viewpoint shifts with fewer holes and artifacts.

## Abstract

Existing depth estimation methods are fundamentally limited to predicting depth on discrete image grids. Such representations restrict their scalability to arbitrary output resolutions and hinder the geometric detail recovery. This paper introduces **InfiniDepth**, which represents depth as neural

implicit fields. Through a simple yet effective local implicit decoder, we can query depth at continuous 2D coordinates, enabling arbitrary-resolution and fine-grained depth estimation. To better assess our method’s capabilities, we curate a high-quality 4K synthetic benchmark from five different games, spanning diverse scenes with rich geometric and appearance details. Extensive experiments demonstrate that InfiniDepth achieves state-of-the-art performance on both synthetic and real-world benchmarks across relative

\*Equal contribution. †Corresponding author.

and metric depth estimation tasks, particularly excelling in fine-detail regions. It also benefits the task of novel view synthesis under large viewpoint shifts, producing high-quality results with fewer holes and artifacts.

## 1. Introduction

Monocular depth estimation (MDE) is a fundamental task in computer vision, with widespread applications in autonomous driving and robotics. Some traditional methods [23, 24] represent the depth map as a graph-structured output with conditional random fields (CRFs), showing some success in the early stages but are limited in scalability and detail prediction due to optimization complexity.

With the development of deep learning, mainstream depth estimation methods [3, 18, 29, 41, 46, 48, 49] adopt regular 2D grids to represent depth maps, as this representation is naturally compatible with modern neural network architectures. Although these methods demonstrate strong generalization, they struggle to produce high-resolution depth maps while preserving fine details, and tend to fail to accurately predict depth in regions with significant geometric variations. Fundamentally, these limitations stem from the discrete grid-based depth representation, which constrains depth prediction at fixed grid locations, inherently limiting output resolution to the training image size. Moreover, predicting depth on entire grids requires either convolutional upsampling or linear projection from latents to depth patches. The former introduces smoothing effects, while the latter struggles to capture local geometric variations—both sacrificing high-frequency details.

In this paper, we present **InfiniDepth**, a new depth representation that models depth as neural implicit fields, enabling arbitrary-resolution and fine-grained depth estimation. Specifically, an input image is encoded by a vision transformer into multi-stage feature tokens, followed by a reassemble block that constructs a feature pyramid. Then, for any continuous 2D coordinate  $(x, y)$ , we gather spatially aligned features from the pyramid within a local window and feed them into a lightweight MLP to predict depth. Unlike prior methods constrained to grid-based depth prediction, InfiniDepth adopts a continuous and localized prediction paradigm. It is no longer constrained by training resolutions and naturally produces arbitrary-resolution depth maps with fine details (Fig. 1 (a)). The localized prediction further excels at capturing geometric variations, producing fine-grained point clouds (Fig. 1 (b)).

Another benefit of InfiniDepth is its ability to enhance novel view synthesis (NVS) quality under large viewpoint shifts. Specifically, recent feed-forward NVS methods [35, 47] predict pixel-aligned depth and Gaussian parameters. Unprojecting such a discrete per-pixel depth map produces a surface point cloud with strong density imbalance due to

perspective projection and surface orientation, thereby degrading NVS quality under large viewpoint shifts. To address this limitation, we design a depth query strategy that allocates sub-pixel query budgets proportionally to each pixel’s corresponding 3D surface element, producing spatially uniform 3D points on object surfaces. With the uniform 3D points, our method produces high-quality novel view synthesis, with markedly fewer holes and reduced artifacts under large viewpoint shifts (Fig. 1 (c)).

To better assess resolution scalability and detail prediction capabilities, we curate Synth4K, a high-quality benchmark collected from five different games, covering diverse scenes with 4K ground-truth depth maps. We also construct high-frequency depth masks to isolate fine-detail regions for targeted evaluation of detail prediction. Extensive experiments on Synth4K and real-world benchmarks demonstrate that InfiniDepth consistently achieves state-of-the-art performance across both relative and metric depth estimation tasks, with particularly strong results in fine-detail regions. Furthermore, we demonstrate that InfiniDepth combined with a depth query strategy can benefit novel view synthesis under large viewpoint shifts.

In summary, this work has the following contributions:

- We propose a new depth representation that models depth as neural implicit fields and demonstrate its capability for arbitrary-resolution and fine-grained depth estimation.
- We design a depth query strategy that produces uniformly distributed 3D points on object surfaces, improving novel view synthesis quality under large viewpoint shifts.
- We curate Synth4K, a high-quality 4K benchmark for evaluating depth estimation methods at high resolution and fine geometric details.

## 2. Related Work

**Relative Depth Estimation.** Relative depth estimation aims to infer a normalized depth map without absolute scale. Recent works [29, 41, 48, 49] adopt Vision Transformer (ViT) backbones [7] with convolutional decoders to regress 2D discretized depth maps. DepthAnything [48, 49] improves generalization by combining labeled and large-scale unlabeled data, while MoGe [41] enhances geometric accuracy with affine-invariant point maps and optimal training supervision. Diffusion-based methods [11, 13, 18, 46] model the distribution of depth maps, with Marigold [18] leveraging pretrained diffusion priors and PPD [46] refining depth boundaries via a semantics-prompted DiT. However, all these methods represent depth as discrete 2D grids, limiting resolution scalability and fine detail recovery.

**Metric Depth Estimation.** Early metric depth methods [1, 2, 8, 21] typically formulate the problem as a global distribution classification task or fine-tune depth models on datasets with metric depth annotations. Recent ap-

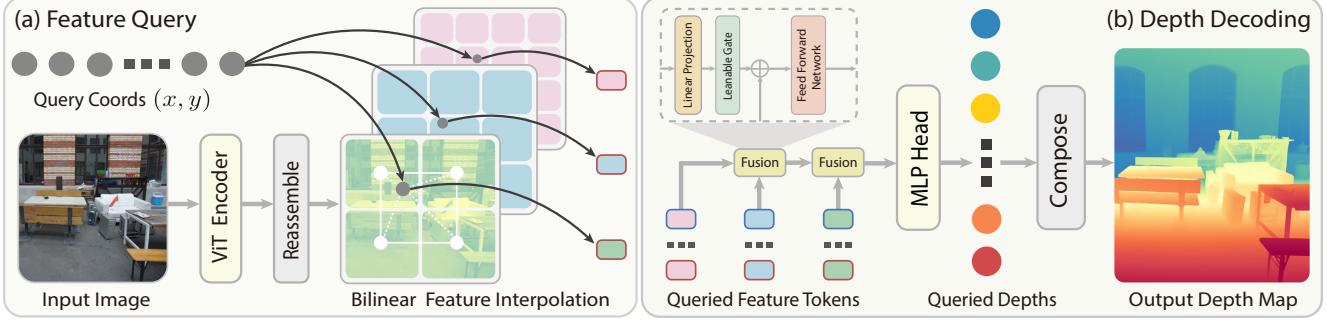


Figure 2. **Overview of InfiniDepth.** (a) Feature Query: given an input image and a continuous query 2D coordinate, we extract feature tokens from multiple layers of the ViT encoder, and query local features for the coordinate at each scale through bilinear interpolation. (b) Depth Decoding: given the multi-scale local features queried at the continuous coordinate, we hierarchically fuse features from high spatial resolution to low spatial resolution, and decode the fused feature to the depth value through a MLP head.

proaches [3, 14, 28, 51, 52] address the ambiguity by incorporating camera intrinsics, while others [22, 25, 39, 44, 53] leverage sparse depth as additional inputs to improve accuracy. For example, PriorDA [44] and Omni-DC [53] complete depth maps under various patterns of sparse depth inputs, while Marigold-DC [39] parameterizes the scale and shift of metric depth and optimizes them iteratively. PromptDA [22] introduces a novel depth prompt module for accurate estimation, but fine-grained geometry recovery remains challenging. In this work, we demonstrate that InfiniDepth combined with sparse depth inputs can significantly enhance metric depth estimation, especially in predicting fine-grained geometry details.

**Implicit Neural Representations.** *Implicit Neural Representations* (INRs) map continuous coordinates to signals and have been widely applied in 3D reconstruction and beyond. NeRF [26] models scenes as neural radiance fields and PifU [31] uses a pixel-aligned implicit function to relate image pixels to 3D human geometry. The paradigm has also been extended to 2D images, optical flow, and multi-view scene representation. LIIF [5] learns continuous image representation with an implicit function and AnyFlow [16] achieves arbitrary scale optical flow with implicit representation. DeFiNe [12] proposes an implicit multi-view scene representation, but architectural constraints limit it to low-resolution outputs. Inspired by these advances, we represent depth as neural implicit fields along with a simple yet effective implicit decoder, enabling arbitrary-resolution and fine-grained depth estimation.

### 3. Method

Given a single RGB image, our goal is to estimate depth for any continuous 2D coordinate in the image plane. The overview of our method is illustrated in Fig. 2.

#### 3.1. Representing Depth as Neural Implicit Fields

Neural implicit fields model signals  $y$  as an implicit function of the continuous coordinates  $x$  parameterized by a neural network:

$$y = F_\theta(x), \quad (1)$$

where  $F_\theta$  is typically implemented as a multi-layer perceptron (MLP). Compared to explicit representations such as voxels or image grids whose fidelity scales with discretization, neural implicit field models fine-grained geometry in a resolution-agnostic manner with fewer parameters.

We extend the concept of neural implicit fields to represent depth, which models depth estimation as an implicit function that maps any continuous 2D coordinate  $(x, y) \in [0, W] \times [0, H]$  to depth value  $d_I(x, y)$ , conditioned on the input RGB image  $I \in \mathbb{R}^{H \times W \times 3}$ :

$$d_I(x, y) = N_\theta(I, (x, y)), \quad (2)$$

where  $N_\theta$  is parameterized by a neural network.

#### 3.2. Multi-Scale Local Implicit Decoder

We instantiate  $N_\theta$  in Eq. 2 as a multi-scale local implicit decoder, which reassembles and aggregates features from multiple layers of the image encoder for any continuous query coordinate  $(x, y)$ , with a lightweight MLP head to predict depth values. This simple yet effective decoder consists of two modules: **Feature Query** and **Depth Decoding**.

**Feature Query.** The input image  $I$  is firstly encoded by a Vision Transformer to obtain a set of feature tokens. Following [29], we design a reassemble block which extracts feature tokens from multiple ViT layers and projects them to different hidden dimensions. To capture fine local details and preserve global semantics, we upsample shallow-layer features (pink and blue tokens in Fig. 2 (a)) to higher spatial resolutions, while retaining deeper-layer features (green tokens) at their native resolution. In this way, we construct a feature pyramid  $\{f^k\}_{k=1}^L$  with  $f^k \in \mathbb{R}^{h_k \times w_k \times C^k}$ .

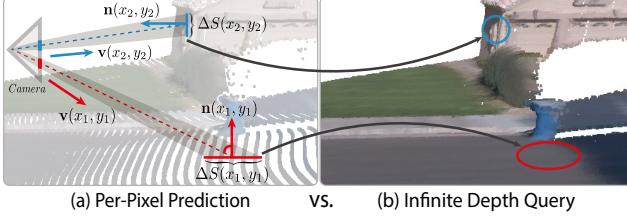


Figure 3. **Advantage of Our Infinite Depth Query.** The blue and red highlighted regions represent areas with different depths, surface normals, and viewing directions. Per-pixel depth prediction leads to strong density imbalance in these regions due to perspective projection and surface orientation, while Infinite Depth Query applies sub-pixel query with adaptive weights to generate uniformly distributed point clouds.

For a continuous query coordinate  $(x, y) \in [0, W] \times [0, H]$  in the image plane, we first map it to the corresponding location  $(x_k, y_k) \in [0, w_k] \times [0, h_k]$  at the  $k$ -th feature-pyramid scale:

$$(x_k, y_k) = \left( x \cdot \frac{w_k}{W}, y \cdot \frac{h_k}{H} \right).$$

For each scale  $k$ , we define the local grid neighborhood around  $(x_k, y_k)$  as  $\mathcal{N}_k(x_k, y_k)$ , which is  $\{(i, j) \mid i \in \{\lfloor x_k \rfloor, \lfloor x_k \rfloor + 1\}, j \in \{\lfloor y_k \rfloor, \lfloor y_k \rfloor + 1\}\}$ , and aggregate features from this neighborhood using bilinear interpolation, yielding a feature token  $f_{(x,y)}^k \in \mathbb{R}^{1 \times C^k}$  for the query coordinate  $(x, y)$  at scale  $k$ .

**Depth Decoding.** Given the multi-scale local descriptors  $\{f_{(x,y)}^k\}_{k=1}^L$ , we fuse them hierarchically from shallow (detail) to deep (semantic) features, aiming to better capture fine-grained geometric variations, preserve both local details and global context, and achieve high-precision and robust depth decoding.

Let  $\mathbf{h}_1 := f_{(x,y)}^1 \in \mathbb{R}^{C_1}$  denote the queried feature at the shallowest (highest-resolution) scale. For each scale  $k = 1, \dots, L-1$ , we hierarchically fuse  $\mathbf{h}_k$  with the next-scale feature  $f_{(x,y)}^{k+1} \in \mathbb{R}^{C_{k+1}}$  using a residual gated fusion block:

$$\mathbf{h}_{k+1} = \text{FFN}_k \left( f_{(x,y)}^{k+1} + \mathbf{g}_k \odot \text{Linear}(\mathbf{h}_k) \right), \quad (3)$$

where  $\text{Linear}(\cdot)$  denotes a linear projection to match the feature dimension,  $\mathbf{g}_k \in (0, 1)^{C_{k+1}}$  is a learnable channel-wise gate, and  $\odot$  denotes element-wise multiplication. Here,  $\text{FFN}_k(\cdot)$  denotes a two-layer feed-forward network with non-linear activation. This process is repeated from  $k = 1$  to  $L-1$ , resulting in the final fused feature  $\mathbf{h}_L \in \mathbb{R}^{C_L}$  at the deepest scale. Finally, the depth value at  $(x, y)$  is predicted by an MLP head:

$$d_I(x, y) = \text{MLP}(\mathbf{h}_L). \quad (4)$$

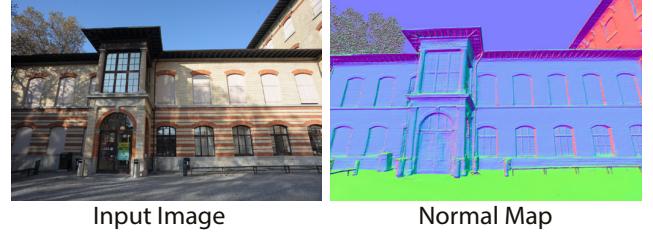


Figure 4. **Normal map from implicit fields through *torch autograd***, indicating high-quality internal geometry of our model.

### 3.3. Infinite Depth Query

Unprojecting a discrete per-pixel depth map produces a surface point cloud with strong density imbalance due to perspective projection and surface orientation (Fig. 3 (a)), thereby degrading NVS quality under large viewpoint shifts. We present a depth query strategy that generates approximately uniform 3D points on visible surfaces by leveraging our implicit depth field.

The key insight is that the 3D surface area  $\Delta S(x, y)$  corresponding to each pixel depends on two geometric factors: (1) **depth-squared scaling**—pixels at greater depth cover a surface area that grows quadratically with distance ( $\propto d^2$ ), and (2) **surface orientation effect**—when the surface normal deviates from the viewing direction, its projection onto the image is compressed, causing each pixel to cover a larger actual surface area. We counteract these effects by allocating sub-pixel query budgets proportionally to each pixel’s corresponding 3D surface element.

Specifically, we first query depth at pixel coordinates  $(x, y)$  and back-project them to 3D points  $\mathbf{X}(x, y)$ , then derive an adaptive weight  $w(x, y)$  that estimates the differential surface area  $\Delta S(x, y)$  at each pixel location:

$$w(x, y) = \frac{d_I(x, y)^2}{|\mathbf{n}(x, y) \cdot \mathbf{v}(x, y)| + \varepsilon} \propto \Delta S(x, y), \quad (5)$$

where  $d_I(x, y)$  denotes the queried depth,  $\mathbf{n}(x, y)$  is the surface normal,  $\mathbf{v}(x, y)$  represents the unit viewing direction, and  $\varepsilon$  is a small constant for numerical stability. In this formulation,  $d_I(x, y)^2$  accounts for depth-squared scaling, while  $|\mathbf{n}(x, y) \cdot \mathbf{v}(x, y)|$  compensates for surface orientation effect, together approximating the 3D surface area subtended by each pixel.  $\mathbf{n}(x, y)$  is computed from the Jacobian of  $\mathbf{X}(x, y)$  with respect to continuous image coordinates, leveraging the differentiable nature of our implicit depth field (Fig. 4):

$$\mathbf{n}(x, y) = \frac{\partial_x \mathbf{X}(x, y) \times \partial_y \mathbf{X}(x, y)}{\|\partial_x \mathbf{X}(x, y) \times \partial_y \mathbf{X}(x, y)\|} \in \mathbb{R}^3. \quad (6)$$

Based on  $w(x, y)$ , we allocate adaptive query budgets and uniformly distribute sub-pixel query coordinates within

each pixel patch. Querying  $d_I(x, y)$  at these continuous coordinates and back-projecting to 3D yields a point cloud with approximately uniform surface coverage (Fig. 3 (b)).

See *supp.* for implementation details and visualizations.

### 3.4. Implementation Details

**Network Architecture.** We adopt the DINOv3 [34] ViT-Large model as our image encoder. We extract feature maps from layers 4, 11, and 23 of the encoder and project them to hidden dimensions of 256, 512, and 1024, respectively. The feature maps from layers 4 and 11 are then upsampled by factors of 4 and 2, respectively. See *supp.* for more details on the network design, as well as an evaluation of computational efficiency and parameter count.

**Training Data and Strategies.** Given our goal of achieving fine-grained depth estimation, we exclusively train our model on synthetic datasets, as real-world datasets often contain noisy and incomplete depth maps. We utilize Hypersim [30], VKITTI [4], TartanAir [43], IRS [40], etc., along with several high-resolution datasets including UnrealStereo4K [37] and UrbanSyn [10].

Due to the properties of our depth representation, we can flexibly supervise sparse samples instead of the entire depth map. Specifically, we randomly draw  $N$  coordinate-depth pairs and compute the  $l1$  loss over these points:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N |d_i - \hat{d}_i|, \quad (7)$$

where  $d_i$  is the ground truth, and  $\hat{d}_i$  is the predicted depth. We train our model using the AdamW optimizer with a learning rate of  $1 \times 10^{-5}$ . It's trained for 800k steps using 8 NVIDIA GPUs, with a batch size of 4 per GPU. See *supp.* for more details on training data and strategies.

## 4. Experiments

### 4.1. Synth4K

To assess zero-shot generalization, prior methods are commonly evaluated on real-world benchmarks. However, ground-truth depth in these datasets is typically low-resolution and sparse, often failing to capture fine geometric structures such as edges and high-frequency details. This makes it challenging to reliably evaluate models on high-resolution and fine-grained depth estimation.

To address this limitation, we curate a high-quality synthetic benchmark named Synth4K, specifically designed for zero-shot evaluation. Synth4K consists of RGB-D data collected from five different games (denoted as Synth4K-1 to Synth4K-5). Each subset contains hundreds of 4K-resolution image pairs, spanning diverse indoor and outdoor environments. We further compute a multi-scale Laplacian

energy map for each depth image and sample pixels proportionally to the energy to construct a binary high-frequency (HF) mask. This mask highlights geometrically detailed regions and enables targeted evaluation. See *supp.* for more implementation details and visualizations of Synth4K.

Compared with existing benchmarks, Synth4K provides significantly higher depth-map resolution and substantially improved detail coverage, establishing a stronger benchmark for high-resolution and fine-grained depth estimation.

### 4.2. Experimental Setup

**Relative Depth Estimation.** Following prior work, we evaluate zero-shot relative depth estimation on five real-world datasets: KITTI [9], ETH3D [32], NYUv2 [33], ScanNet [6], and DIODE [38]. The  $\delta_1$  accuracy is reported in Table 3. To demonstrate our method's capability for arbitrary-resolution and fine-grained depth estimation, we further conduct extensive experiments on Synth4K, including evaluating depth predictions over the full 4K-resolution images, as well as assessing fine-detail prediction performance in HF-masked regions. We report  $\delta_{0.5}$ ,  $\delta_1$ , and  $\delta_2$  in Table 1.  $\delta_{0.5}$ ,  $\delta_1$ , and  $\delta_2$  denote the percentage of pixels satisfying  $\max\left(\frac{d}{d^*}, \frac{d^*}{d}\right) < 1.25^{0.5}$ ,  $1.25^1$ , and  $1.25^2$ , respectively, where  $d$  is the predicted depth and  $d^*$  is the ground-truth depth.

**Metric Depth Estimation.** To demonstrate that InfiniDepth is also effective for metric depth estimation, we incorporate sparse depth inputs using the depth prompt module proposed in [22], referred to as **Ours-Metric** for clarity. We report  $\delta_{0.01}$ ,  $\delta_{0.02}$ ,  $\delta_{0.04}$  accuracy in Table 2 and Table 4.  $\delta_{0.01}$ ,  $\delta_{0.02}$ ,  $\delta_{0.04}$  denote the percentage of pixels where  $\max\left(\frac{d}{d^*}, \frac{d^*}{d}\right) < 1.01$ ,  $1.02$ ,  $1.04$ , respectively. These stricter thresholds are adopted because metric depth estimation with sparse depth inputs typically achieves higher prediction accuracy and therefore warrants more stringent evaluation criteria.

### 4.3. Comparisons with the State of the Art

We compare our approach to two categories of SOTA methods on both Synth4K and real-world benchmarks: (1) relative depth estimation using only RGB inputs, and (2) metric depth estimation with additional sparse depth. For relative depth estimation, we evaluate against DepthAnything [48], DepthAnythingV2 [49], DepthPro [3], MoGe [41], MoGe2 [42], Marigold [39], and PPD [46], aligning predictions to ground-truth depth before evaluation. For metric depth estimation, we compare with depth completion approaches, including Marigold-DC [39], Omni-DC [53], PriorDA [44], and PromptDA [22]. To ensure fair comparisons, we use the same input resolution across all baselines and the same sparse depth samples for

Type	Method	Synth4K-1			Synth4K-2			Synth4K-3			Synth4K-4			Synth4K-5		
		$\delta_{0.5}$	$\delta_1$	$\delta_2$												
Full Image	DepthAnything [48]	70.4	83.8	93.0	77.9	88.2	95.2	77.4	88.6	96.0	83.9	92.8	96.6	84.3	93.0	97.4
	DepthAnythingv2 [49]	67.3	81.3	91.0	76.0	88.1	95.4	71.4	85.5	95.3	86.1	94.1	97.4	78.6	92.1	97.6
	DepthPro [3]	63.5	80.2	91.2	66.7	83.1	93.5	61.2	80.2	92.1	87.1	94.1	97.1	73.9	89.1	96.7
	MoGe [41]	69.3	83.7	92.7	72.8	86.2	94.1	70.6	85.6	94.0	89.2	94.6	97.0	81.1	92.7	97.7
	MoGe-2 [42]	69.0	84.2	93.4	73.5	86.6	94.3	70.9	85.3	94.0	90.4	95.3	97.6	80.7	92.4	97.9
	Marigold [39]	54.6	72.9	85.1	57.2	75.6	87.8	55.6	73.7	85.7	79.3	90.7	95.6	66.5	84.5	93.3
	PPD [46]	61.5	81.1	92.5	62.2	84.6	93.9	57.5	82.8	93.9	85.6	94.1	97.0	69.1	90.4	96.5
High-Freq Details	<b>Ours</b>	74.3	89.0	96.1	80.4	92.2	97.0	82.0	93.9	97.8	89.7	95.5	98.0	88.5	96.3	98.8
	DepthAnything [48]	43.4	61.3	78.3	41.0	59.4	77.4	44.3	62.1	80.2	55.1	70.3	82.0	53.1	70.8	86.0
	DepthAnythingv2 [49]	43.0	60.6	77.9	41.4	60.1	78.2	41.8	60.7	80.0	59.3	73.9	84.7	49.2	70.3	86.6
	DepthPro [3]	43.4	62.4	80.6	38.4	58.8	79.3	38.2	58.6	79.6	62.6	76.1	85.3	53.3	73.1	89.0
	MoGe [41]	48.8	65.8	80.9	43.9	61.6	77.9	45.9	62.9	79.4	64.4	75.7	83.9	60.6	76.2	88.5
	MoGe-2 [42]	48.9	66.5	82.6	44.3	62.5	79.3	46.0	63.4	80.6	66.7	78.2	85.8	61.4	77.3	89.4
	Marigold [39]	35.8	54.0	72.0	30.8	49.4	69.9	33.4	51.4	71.1	54.2	69.9	81.2	43.9	63.2	81.1
<b>Ours-Metric</b>	PPD [46]	42.3	61.6	79.6	36.6	58.3	77.8	36.9	58.5	78.0	61.6	75.3	84.4	48.3	70.1	86.3
	<b>Ours</b>	49.2	67.5	83.1	46.7	65.6	81.9	52.5	69.0	83.1	65.3	78.2	87.3	63.9	79.5	90.7

Table 1. **Zero-shot relative depth estimation on Synth4K.** The top-3 results are highlighted as first , second , and third .

Type	Method	Synth4K-1			Synth4K-2			Synth4K-3			Synth4K-4			Synth4K-5		
		$\delta_{0.01}$	$\delta_{0.02}$	$\delta_{0.04}$												
Full Image	Marigold-DC [39]	19.5	31.9	48.0	13.2	22.4	36.1	18.6	32.1	49.5	26.9	40.5	54.1	18.0	31.4	49.0
	Omni-DC [53]	38.8	46.0	54.1	38.4	43.8	52.5	44.0	49.5	55.1	37.9	43.2	58.9	43.4	50.5	55.7
	PriorDA [44]	44.8	67.2	80.7	47.3	67.9	78.6	55.5	75.4	85.0	61.9	78.4	88.0	54.0	75.9	86.9
	PromptDA [22]	65.0	79.8	88.0	66.3	78.1	85.4	72.0	84.8	90.8	78.8	88.6	93.1	69.2	84.8	91.2
	<b>Ours-Metric</b>	78.0	86.7	92.0	76.6	83.6	89.0	83.8	90.1	93.5	87.2	92.0	95.0	83.1	89.8	93.5
High-Freq Details	Marigold-DC [39]	9.0	15.8	26.0	5.3	9.7	17.2	8.3	15.0	24.9	13.4	22.5	34.1	10.3	18.8	31.8
	Omni-DC [53]	18.4	26.4	36.3	12.4	19.0	28.4	22.9	30.3	37.9	21.8	30.1	42.1	24.1	34.1	44.7
	PriorDA [44]	12.6	21.7	33.7	8.5	15.1	24.9	13.4	21.7	31.5	20.2	31.9	45.8	19.0	31.8	45.6
	PromptDA [22]	21.1	33.1	45.7	15.3	24.5	36.6	24.7	35.3	45.3	32.0	45.2	57.2	27.3	41.4	54.0
	<b>Ours-Metric</b>	33.2	46.5	58.7	24.0	34.9	47.8	37.2	47.6	56.5	45.5	57.5	68.2	38.8	52.0	63.5

Table 2. **Zero-shot metric depth estimation on Synth4K.** The top-3 results are highlighted as first , second , and third .

Method	KITTI	ETH3D	NYUv2	ScanNet	DIODE
	$\delta_1$	$\delta_1$	$\delta_1$	$\delta_1$	$\delta_1$
DepthAnything [48]	97.5	98.4	97.8	97.8	97.3
DepthAnythingV2 [49]	96.7	97.8	97.3	97.4	97.0
DepthPro [3]	97.5	98.0	97.6	97.9	97.1
MoGe [41]	98.3	98.9	98.0	98.2	97.4
MoGe-2 [42]	98.3	99.0	98.2	98.4	97.4
Marigold [39]	94.2	96.8	95.8	93.9	94.7
PPD [46]	97.3	98.3	97.2	97.3	96.2
<b>Ours-Relative</b>	97.9	99.1	97.6	97.3	97.4

Table 3. **Zero-shot relative depth estimation on real-world datasets.** The top-3 results are highlighted.

metric methods. On Synth4K, baseline outputs are upsampled to 4K, whereas InfiniDepth is queried directly at 4K.

As shown in Table 1 and Table 2, our method significantly outperforms all existing methods across all metrics on Synth4K, highlighting its strength in high-resolution and fine-grained depth estimation. On real-world benchmarks, **Ours** performs on par with current SOTA methods, while **Ours-Metric** achieves clear improvements over existing metric depth estimation methods, as shown in Table 3

Method	KITTI	ETH3D	NYUv2	ScanNet	DIODE
	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$
Marigold-DC [39]	36.6	72.6	71.4	76.7	84.2
Omni-DC [53]	17.5	57.4	62.1	55.8	83.3
PriorDA [44]	54.1	85.7	78.1	82.7	94.3
PromptDA [22]	58.3	92.8	83.6	87.0	97.3
<b>Ours-Metric</b>	63.9	96.7	86.9	90.4	98.4

Table 4. **Zero-shot metric depth estimation on real-world datasets.** The top-3 results are highlighted.

and Table 4. (Marigold-DC suffers from VAE-based quantization loss, as discussed in [46], leading to low metric accuracy.) Qualitative depth map (Fig. 5) and point cloud comparisons further illustrate the advantages of our approach in producing accurate and detailed predictions.

#### 4.4. Ablations and Analysis

**Depth Representation.** To verify the effectiveness of our depth representation, we compare it with a baseline that predicts depth on discrete grids using a DPT decoder [29]. Both models share the same encoder (DINOv3 ViT-Large)

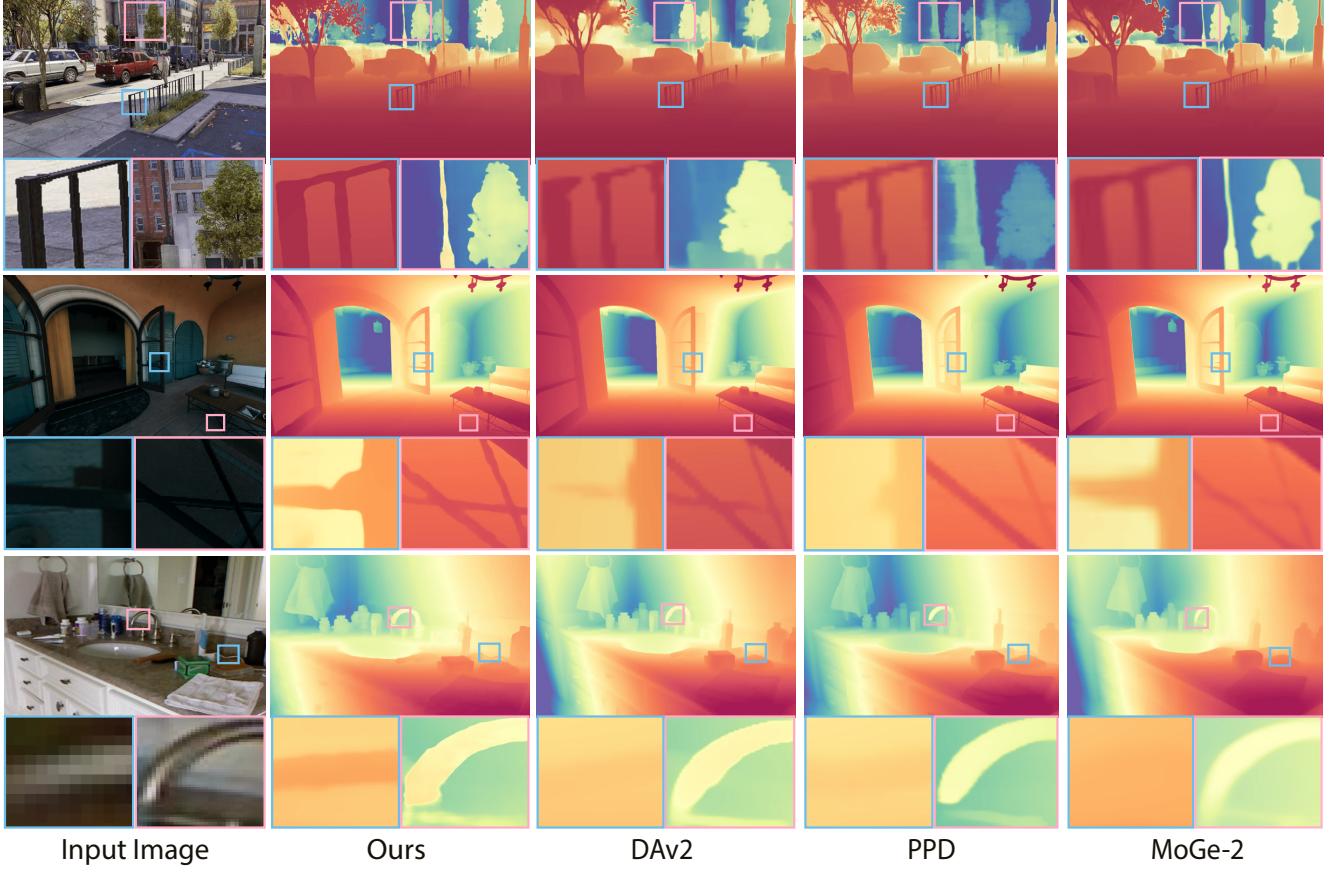


Figure 5. **Qualitative comparisons for relative depth estimation.** The first two rows show prediction results on Synth4K, while the last row shows real-world data with low resolution RGB inputs. The boxes highlight detail regions upsampled to higher resolution for baselines, while our method directly predicts at this resolution. More comparisons can be found in the supp..

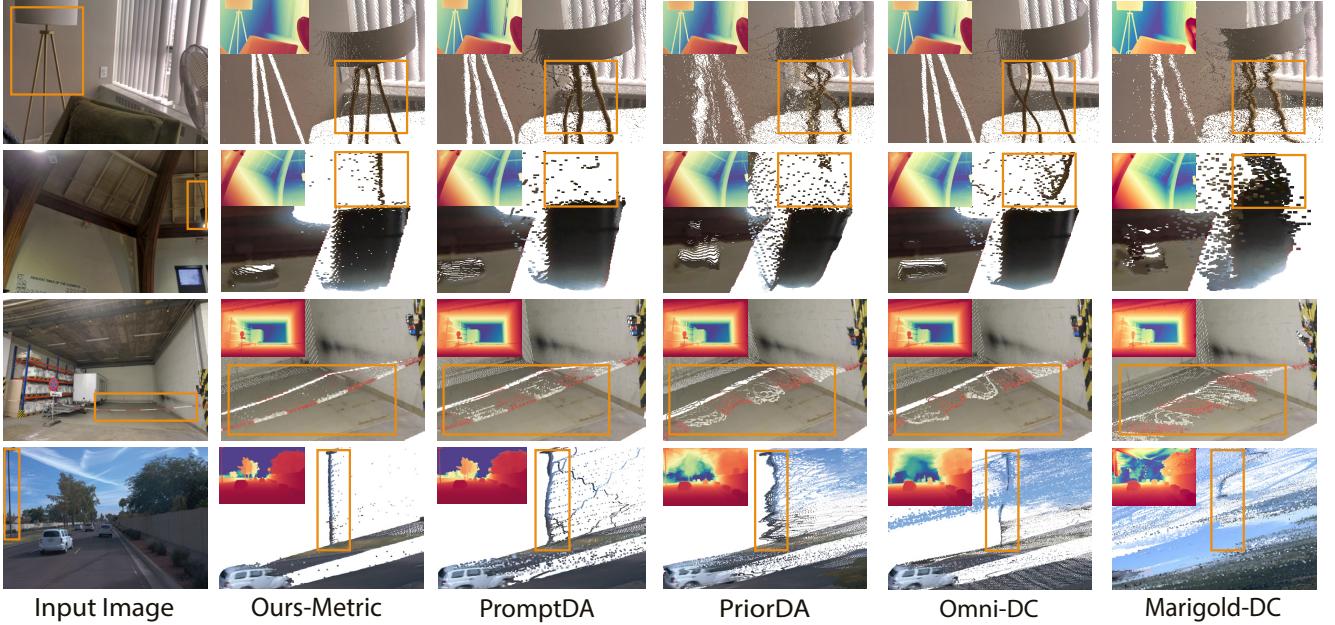


Figure 6. **Qualitative comparisons for metric depth estimation.** The boxes highlight the high-frequency geometric details.

Ablation	Synth4K-1	Synth4K-2	Synth4K-3	Synth4K-4	Synth4K-5	KITTI	ETH3D	NYUv2	ScanNet	DIODE
	$\delta_{0.01}$									
<b>Full Model</b>	<b>72.7</b>	<b>73.5</b>	<b>78.2</b>	<b>81.5</b>	<b>79.4</b>	<b>61.7</b>	<b>93.9</b>	<b>84.7</b>	<b>88.5</b>	<b>97.6</b>
w/o Neural Implicit Fields	62.4	65.1	66.5	73.2	68.9	49.0	88.9	81.2	84.2	95.4
w/o Multi-Scale Query	<u>66.6</u>	<u>67.4</u>	<u>70.8</u>	77.0	<u>72.4</u>	<u>59.7</u>	88.7	<u>82.5</u>	<u>86.2</u>	95.6
w/o DINOv3 [34]	63.8	66.2	67.9	<u>77.0</u>	71.7	57.9	<u>90.1</u>	80.8	83.2	<u>95.8</u>

Table 5. **Quantitative ablations on different datasets.** The **best** and **second best** are highlighted. See Sec. 4.4 for details.

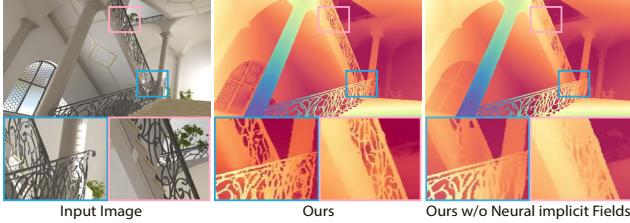


Figure 7. **Qualitative ablations on depth representation.** The boxes highlight the fine-detail regions.

and training data (Hypersim). The quantitative results show that representing depth as neural implicit fields yields substantially better performance for metric depth estimation (Table 5), with moderate gains for relative depth estimation (See *supp.*). This gap is expected. Sparse depth inputs greatly reduce the ambiguity of metric depth estimation, allowing more convincing and consistent results. Our depth representation—together with its localized prediction mechanism—further enhances depth precision, yielding clear improvements in both quantitative metrics and visual quality. In contrast, RGB-only relative depth estimation suffers from high depth ambiguity, causing quantitative metrics to saturate. Nevertheless, our representation consistently recovers finer geometric details, as shown in Fig. 7.

**Design Choices for Implicit Decoder.** We ablate the multi-scale feature query and fusion mechanism in our implicit decoder, against a baseline that samples features only from the single-scale final feature map of the image encoder for each query coordinate. The quantitative ablation results in Table 5 demonstrate that multi-scale feature query mechanism brings significant improvements across datasets. We also compare more detailed design choices in *supp.*, including (1) explicitly learning offsets between query coordinates and grid neighborhood vs. bilinear interpolation, (2) employing a cross-attention mechanism for feature querying at given coordinates vs. a shared MLP, etc.

**Image Encoder.** We investigate the impact of different image encoders in our framework by comparing DI-NOv3 [34] and DI-NOv2 [27], both using ViT-Large backbones. Quantitative results are summarized in Table 5.

See *supp.* for more ablations and analysis of our method.

## 4.5. Application: Single-View Novel View Synthesis

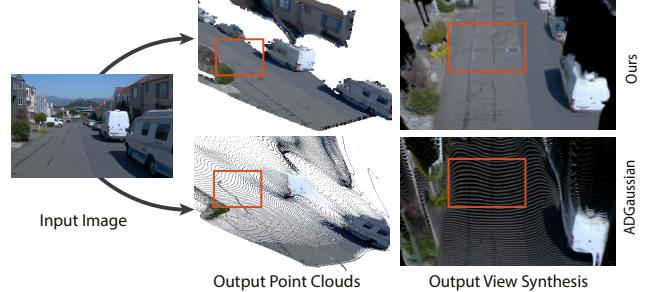


Figure 8. **NVS driven by InfiniDepth and ADGaussian [35].** The boxes highlight regions with geometric holes in the ADGaussian predictions. Refer to *supp.* for more results.

We demonstrate that InfiniDepth combined with the depth query strategy (Sec. 3.3) significantly improves single-view novel view synthesis (NVS) under large viewpoint shifts.

Specifically, we extend our depth model with a lightweight Gaussian Splatting (GS) head. See *supp.* for more details on the GS head design and training. At inference time, we first apply the depth query strategy to generate uniformly distributed points on visible surfaces, which serve as Gaussian centers, then feed them to the GS head to predict Gaussian attributes and render novel views under large viewpoint shifts. As shown in Fig. 1 (c) and Fig. 8, ADGaussian [35], which predicts pixel-aligned depth, often exhibits noticeable geometric holes and artifacts. In contrast, InfiniDepth produces more complete and stable novel view synthesis results even under large viewpoint shifts.

## 5. Conclusion and Discussions

This paper presents a new depth representation that models depth as neural implicit fields. This formulation enables depth estimation at arbitrary continuous 2D coordinates while better preserving fine-grained geometric details. The effectiveness of the proposed representation is validated on both Synth4K and real-world benchmarks, across different tasks including relative and metric depth estimation. Our method achieves significant improvements in depth prediction accuracy and detail recovery. Combined with a depth query strategy, it further benefits single-view novel view synthesis under large viewpoint shifts.

**Limitations and Future Work.** This work focuses on monocular depth estimation and is trained only on single-view depth data, so when applied to videos it does not explicitly enforce temporal consistency and may exhibit flickering across frames. Future work includes extending our depth representation to multi-view settings to improve temporal stability and 3D consistency. We hope that InfiniDepth will inspire further research on high-resolution, fine-grained depth estimation and its integration into broader 3D perception and reconstruction pipelines.

## References

- [1] 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. [2](#)
- [2] Shariq Farooq Bhat, Reiner Birk, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. [2](#)
- [3] Aleksei Bochkovskii, Amañgl 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 preprint arXiv:2410.02073*, 2024. [2, 3, 5, 6](#)
- [4] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. [5](#)
- [5] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5939–5948, 2019. [3](#)
- [6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. [5](#)
- [7] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. [2](#)
- [8] 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. [2](#)
- [9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013. [5](#)
- [10] Jose L Gómez, Manuel Silva, Antonio Seoane, Agnès Borrás, Mario Noriega, Germán Ros, Jose A Iglesias-Guitian, and Antonio M López. All for one, and one for all: Urbansyn dataset, the third musketeer of synthetic driving scenes. *Neurocomputing*, 637:130038, 2025. [5](#)
- [11] Ming Gui, Johannes Schusterbauer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Depthfm: Fast generative monocular depth estimation with flow matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3203–3211, 2025. [2](#)
- [12] Vitor Guizilini, Igor Vasiljevic, Jiading Fang, Rare Ambru, Greg Shakhnarovich, Matthew R Walter, and Adrien Gaidon. Depth field networks for generalizable multi-view scene representation. In *European Conference on Computer Vision*, pages 245–262. Springer, 2022. [3](#)
- [13] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124*, 2024. [2](#)
- [14] 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. [3](#)
- [15] 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. [2](#)
- [16] Hyunyoung Jung, Zhuo Hui, Lei Luo, Haitao Yang, Feng Liu, Sungjoo Yoo, Rakesh Ranjan, and Denis Demanolx. Anyflow: Arbitrary scale optical flow with implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5455–5465, 2023. [3](#)
- [17] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13229–13239, 2023. [2](#)
- [18] 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. [2](#)
- [19] Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Practical stereo matching via cascaded recurrent network with adaptive correlation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16263–16272, 2022. [2](#)
- [20] Yixuan Li, Lihan Jiang, Lining 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. [2](#)
- [21] 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. [2](#)

- [22] 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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17070–17080, 2025. 3, 5, 6
- [23] Faya Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5162–5170, 2015. 2
- [24] Faya Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015. 2
- [25] Zhiheng Liu, Ka Leong Cheng, Qiuyu Wang, Shuzhe Wang, Hao Ouyang, Bin Tan, Kai Zhu, Yujun Shen, Qifeng Chen, and Ping Luo. Depthlab: From partial to complete. *arXiv preprint arXiv:2412.18153*, 2024. 3
- [26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3
- [27] 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. 8
- [28] 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*, pages 10106–10116, 2024. 3
- [29] 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. 2, 3, 6
- [30] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atul Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021. 5
- [31] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2304–2314, 2019. 3
- [32] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3260–3269, 2017. 5
- [33] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 5
- [34] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 5, 8
- [35] Qi Song, Chenghong Li, Haotong Lin, Sida Peng, and Rui Huang. Adgaussian: Generalizable gaussian splatting for autonomous driving with multi-modal inputs. *arXiv preprint arXiv:2504.00437*, 2025. 2, 8
- [36] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 3
- [37] 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. 5
- [38] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 5
- [39] Massimiliano Viola, Kevin Qu, Nando Metzger, Bingxin Ke, Alexander Becker, Konrad Schindler, and Anton Obukhov. Marigold-dc: Zero-shot monocular depth completion with guided diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5359–5370, 2025. 3, 5, 6, 2
- [40] Qiang Wang, Shizhen Zheng, Qingsong Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. Irs: A large naturalistic indoor robotics stereo dataset to train deep models for disparity and surface normal estimation. *arXiv preprint arXiv:1912.09678*, 2019. 5
- [41] 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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5261–5271, 2025. 2, 5, 6
- [42] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025. 5, 6, 2
- [43] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 5
- [44] Zehan Wang, Siyu Chen, Lihe Yang, Jialei Wang, Ziang Zhang, Hengshuang Zhao, and Zhou Zhao. Depth anything with any prior. *arXiv preprint arXiv:2505.10565*, 2025. 3, 5, 6

- [45] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5249–5260, 2025. 2
- [46] Gangwei Xu, Haotong Lin, Hongcheng Luo, Xianqi Wang, Jingfeng Yao, Lianghui 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. 2, 5, 6
- [47] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16453–16463, 2025. 2
- [48] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10371–10381, 2024. 2, 5, 6
- [49] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024. 2, 5, 6
- [50] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1790–1799, 2020. 2
- [51] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 204–213, 2021. 3
- [52] 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. 3
- [53] Yiming Zuo, Willow Yang, Zeyu Ma, and Jia Deng. Omnidc: Highly robust depth completion with multiresolution depth integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9287–9297, 2025. 3, 5, 6

# InfiniDepth: Arbitrary-Resolution and Fine-Grained Depth Estimation with Neural Implicit Fields

## Supplementary Material

### A. Method Details

#### A.1. Implicit Decoder.

We provide additional implementation details of the Feed-Forward Network (FFN) and the MLP head in our implicit decoder.

In the FFN, we first expand the input feature dimension by a factor of four, apply a nonlinear activation, and then compress it back to the original dimension. The MLP head consists of three linear layers with ReLU activations. The input dimension is set to 1024, and the hidden dimension is set to 256. We use ELU activation after the final layer to avoid vanishing gradient issues during training.

#### A.2. Infinite Depth Query

In the main paper, we illustrate how to obtain the adaptive weight  $w_i$  for each pixel  $i$ . Here, we describe how to use  $w_i$  to select sub-pixel query coordinates.

Specifically, we normalize  $w_i$  into a probability distribution

$$p_i = \frac{w_i}{\sum_i w_i}. \quad (8)$$

Given this discrete distribution  $\{p_i\}$ , we construct the cumulative distribution function (CDF):

$$\text{CDF}(k) = \sum_{i=1}^k p_i, \quad (9)$$

which is a monotonically increasing function that maps each pixel index  $k$  to the total probability mass of all pixels up to  $k$ .

We then obtain  $N$  samples using a uniformly stratified inverse-transform sampling scheme. Specifically, we generate a set of uniformly spaced target values

$$q_j = \frac{j + 0.5}{N}, \quad j = 0, \dots, N - 1, \quad (10)$$

and for each  $q_j$ , find the smallest index  $k_j$  such that

$$\text{CDF}(k_j) \geq q_j. \quad (11)$$

This yields  $N$  pixel indices  $\{k_j\}$  whose sampling frequency matches the probability distribution  $\{p_i\}$ .

For each selected pixel  $(u, v)$ , we refine the sampling location by adding a random sub-pixel jitter within  $[-0.5, 0.5]$  around the pixel center:

$$(x, y) = (u + 0.5 + \delta_u, v + 0.5 + \delta_v), \quad \delta_u, \delta_v \sim \mathcal{U}(-0.5, 0.5). \quad (12)$$

Finally,  $(x, y)$  is normalized to match the model's coordinate convention.

#### A.3. Gaussian Splatting (GS) Head

Given the uniform 3D points from Infinite Depth Query, we first enrich each point with color and Plücker ray features extracted from the input image. These per-point features are then combined with features from the ViT encoder to form point-wise tokens. Finally, each token is processed through a MLP and fed into multiple independent linear heads to predict Gaussian attributes, including position offsets  $o$ , color offsets  $c$ , scales  $s$ , opacities  $\alpha$ , and rotations  $r$ , enabling 3D Gaussian splatting for novel view synthesis.

#### A.4. Training Strategies

We present more details of depth normalization, training InfiniDepth and GS head.

**Depth Normalization.** Before depth normalization, we first convert the ground-truth depth values to logarithmic space to reduce the variance between different scenes. Then, we get the affine-invariant normalized depth using:

$$d_{norm} = \frac{d_{log} - d_{min}}{d_{max} - d_{min}}, \quad (13)$$

where  $d_{log}$  is the logarithmic depth value, and  $d_{min}$  and  $d_{max}$  are the 2% and 98% quantiles of the depth values in the logarithmic space, respectively.

**Training InfiniDepth.** We resize the RGB image but remain the original resolution of the ground-truth depth map, as our implicit depth representation allows us to supervise depth predictions at continuous coordinates. We construct coordinates-depth pairs on the original ground-truth depth map, and then randomly sample a set of coordinates during training to compute the  $l_1$  loss. In practice, we sample 100k pairs per image.

**Training GS Head.** We initialize the ViT encoder with the pretrained InfiniDepth weights and keep it frozen, training only the GS head. The GS head is optimized with a learning rate of  $1 \times 10^{-4}$ . Supervision combines an  $l_1$  reconstruction loss and a perceptual LPIPS loss, encouraging both accurate low-frequency color reproduction and high-frequency structural fidelity in the rendered novel views.

#### A.5. Computational Efficiency and Parameter Count

We provide more analysis on the computational efficiency and parameter count of our model and other baseline mod-

els, including DepthPro [3], DepthAnythingV2 [49], MoGe-2 [42], Marigold [39], and PPD [46].

As shown in Tab. 6, the decoder in our model has the lowest parameter count among all compared methods. The computational efficiency of our model is slower than DepthAnythingV2 and MoGe-2. However, the convolution decoder used in DepthAnythingV2 and MoGe-2 makes them less effective in capturing fine-grained depth details. Compared with other methods that also target fine-grained depth estimation, such as DepthPro, Marigold, and PPD, our approach offers better computational efficiency and further surpasses them in the level of detail achieved.

## B. Dataset Details

### B.1. Synth4K

**Dataset curation.** Synth4K is curated from five different games, including *CyberPunk 2077*, *Marvel’s Spider-Man 2*, *Miles Morales*, *Dead Island*, and *Watch Dogs* (Denoted as Synth4K-1, Synth4K-2, Synth4K-3, Synth4K-4, and Synth4K-5, respectively). It contains diverse indoor and outdoor scenes with high-quality graphics and realistic lighting effects. We collect in-game RGB images and corresponding depth maps at a resolution of 3840x2160 (4K) using ReShade, which provides access to the game’s rendering pipeline and enables high-quality capture of both color and depth buffers during gameplay.

**Implementation of high-frequency mask.** To identify high-frequency structures in the depth map  $D \in \mathbb{R}^{H \times W}$ , we compute a geometric energy map that emphasizes local curvature and fine-scale variations.

For a set of smoothing scales  $\{s\}$ , we first obtain multi-scale filtered depth maps

$$D_s = \begin{cases} \text{GaussianBlur}(D, \sigma = s), & s > 0, \\ D, & s = 0. \end{cases} \quad (14)$$

For each scale  $s$ , we compute the absolute Laplacian response using the 4-neighborhood stencil

$$\mathcal{L}(D_s) = \left| D_s * \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \right|, \quad (15)$$

and aggregate the multi-scale response via a per-pixel maximum:

$$E(x, y) = \max_s \mathcal{L}(D_s)(x, y). \quad (16)$$

To suppress extreme outliers, we normalize  $E$  using its 98<sup>th</sup> percentile:

$$\hat{E}(x, y) = \min\left(\frac{E(x, y)}{q_{0.98}(E)}, 1\right), \quad (17)$$

Model	Parameters (M)	Computational Efficiency (s/it)
<b>Ours</b>	15	0.16
DepthPro [3]	29	0.19
DepthAnythingv2 [49]	31	0.03
MoGe-2 [42]	22	0.05
Marigold [39]	-	0.39
PPD [46]	-	1.48

Table 6. **Comparison of parameter count and computational efficiency for different decoders.** Parameters represent the number of parameters in the decoder, while computational efficiency refers to the inference time required by the entire model to process a single  $504 \times 672$  image. We don’t report parameters for Marigold and PPD as they are diffusion-based models.

where  $q_{0.98}(E)$  denotes the 98% quantile of  $E$ .

We further apply temperature-based sharpening to control the contrast of the high-frequency response. Given a temperature parameter  $\tau > 0$ , we define the sharpened energy as

$$\tilde{E}(x, y) = \hat{E}(x, y)^{1/\tau}. \quad (18)$$

Lower temperature values ( $\tau < 1$ ) emphasize sharp structures by amplifying large responses, while higher temperatures ( $\tau > 1$ ) yield a flatter distribution.

Finally, we compute the sampling probability for each pixel as

$$p(x, y) = \frac{\tilde{E}(x, y)}{\sum_{x,y} \tilde{E}(x, y)}, \quad (19)$$

and obtain  $n$  high-frequency candidate locations by sampling from the discrete distribution  $\{p(x, y)\}$  using multinomial sampling.

More visualizations about the RGB images, depth maps and high-frequency masks are provided in Fig. 9 and Fig. 10.

## B.2. Training Datasets

Some of our training datasets are introduced in the main paper. Additionally, we also use the following datasets for training: MatrixCity [20], MVS-Synth [15], Blend-edmvs [50], CREStereo [19], FSD [45], and DynamicReplica [17].

## C. Experiments Details

### C.1. Evaluation Protocols

We ensure the fair comparison of all methods by using consistent input resolutions and evaluation protocols.

On real-world benchmark, we resize the input image to  $504 \times 672$  for all methods, and the output depth maps are evaluated on the same resolution as input, while on Synth4K, we resize the input image to  $504 \times 896$  for all

Ablation	Synth4K-1	Synth4K-2	Synth4K-3	Synth4K-4	Synth4K-5	KITTI	ETH3D	NYUv2	ScanNet	DIODE
	$\delta_{0.01}$									
<b>Sub-Pixel Supervision</b>	<b>72.7</b>	<b>73.5</b>	<b>78.2</b>	<b>81.5</b>	<b>79.4</b>	<b>61.7</b>	<b>93.9</b>	<b>84.7</b>	<b>88.5</b>	<b>97.6</b>
Pixel-Wise Supervision	70.0	70.5	74.7	80.6	76.6	58.8	92.5	84.2	88.0	97.2

Table 7. Quantitative ablations on supervision strategies for metric depth estimation.

Ablation	Synth4K-1	Synth4K-2	Synth4K-3	Synth4K-4	Synth4K-5	KITTI	ETH3D	NYUv2	ScanNet	DIODE
	$\delta_{0.01}$									
<b>Full Model</b>	<b>82.5</b>	84.6	84.9	<b>93.5</b>	<b>92.5</b>	<b>95.2</b>	<b>98.7</b>	<b>97.3</b>	<b>97.2</b>	<b>96.6</b>
w/o Neural Implicit Fields	82.4	<b>85.3</b>	<b>85.3</b>	93.4	90.2	94.6	98.3	96.9	97.1	96.1

Table 8. Quantitative ablations on depth representation for relative depth estimation.

Ablation	KITTI	ETH3D	NYUv2	ScanNet	DIODE
	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$
Bilinear Feat Interp	<b>61.7</b>	<b>93.9</b>	<b>84.7</b>	<b>88.5</b>	<b>97.6</b>
Coordinate-Offset MLP	59.3	90.8	80.5	81.6	96.0
Coordinate-Offset MLP (Local Ensemble)	54.1	84.1	78.7	82.1	95.0
Cross-Attention	54.8	88.2	79.7	80.7	96.2

Table 9. Quantitative ablations on different design choices for metric depth estimation.

methods. The baseline outputs are upsampled to 4K resolution using bilinear interpolation, whereas our method is queried directly at 4K due to the implicit representation.

For the task of relative depth estimation, we align the predicted depth to ground-truth depth using scale-and-shift alignment before evaluation. For the task of metric depth estimation, we sample 1500 sparse depth points from the ground-truth depth map as additional input for all methods. No alignment is applied during evaluation.

## C.2. Single-View Novel View Synthesis (NVS)

Single-View Novel View Synthesis (NVS) aims to generate novel views of a scene given a single input image. When the viewpoint changes significantly such as a Bird’s-Eye View (BEV), prior methods often produce noticeable artifacts and holes due to incomplete geometry estimation. We address this challenge by combining our proposed depth representation with a depth query strategy, generating point clouds that uniformly distribute on object surfaces. Using the Gaussian Splatting (GS) Head described in Sec. A.3, we can render novel view images from the input RGB image and the uniform point clouds, which produces high-quality results with fewer artifacts and holes. We train the GS head on a subset of the Waymo [36] training split and evaluate it on unseen scenes. Qualitative results are shown in Fig. 13.

## C.3. More Ablation Studies

**Supervision strategies.** We ablate different supervision strategies for training our metric depth model, including sub-pixel supervision and pixel-wise supervision. Sub-pixel

supervision refers to using ground-truth depth maps at a higher resolution than the input image during training. This allows us to supervise depth predictions at sub-pixel coordinates within each pixel, which is applied in our full model. Pixel-wise supervision downsamples the ground-truth depth maps to the same resolution of the input image, only providing supervision at the pixel centers. Ablation results in Tab. 7 demonstrate that sub-pixel supervision further improves depth prediction accuracy. It better leverages the inherent property of implicit depth fields to predict depth at continuous coordinates, thereby enhancing the model’s ability for fine-grained depth prediction.

**Depth representation.** We additionally provide quantitative results of different depth representations for relative depth estimation. Results are shown in Tab. 8. Although the metric accuracy does not improve significantly with neural implicit fields, the visual quality of depth maps is noticeably enhanced, as shown in the main paper.

**Design choices of implicit decoder.** Here, we present some different design choices of the feature query module in our implicit decoder, including (1) Coordinate-Offset MLP, (2) Coordinate-Offset MLP (Local Ensemble) and (3) Cross-Attention. Specifically, for (1), we compute the relative offset between a query coordinate and its nearest grid point, and feed the offset into a shared MLP to learn the input coordinate. We then concat the learned coordinate with the feature of the nearest grid point as the queried feature. For (2), we compute the relative offsets between a query coordinate and its four surrounding grid points, and then perform similar operations as (1). For (3), we use the input coordinate as the  $Q$ , and the features of its four surrounding grid points as  $K$  and  $V$ s to perform cross-attention for feature aggregation. We compare the above designs with our default design, which directly uses bilinear interpolation for feature query. Experiments are conducted for metric depth estimation. As shown in Tab. 9, bilinear feature interpola-



Figure 9. **RGB images, depth maps and high-frequency masks in Synth4K.** Each row from top to bottom shows samples from Synth4K’s five games: *CyberPunk 2077*, *Marvel’s Spider-Man 2*, *Miles Morales*, *Dead Island*, and *Watch Dogs*.

tion on feature pyramids achieves the best performance with the least computational cost, while other designs introduce extra parameters and computations but do not lead to performance gains. We also conduct ablations on different image encoders (DINOv2 vs. DINOv3) for our relative depth model, but observe no significant performance differences.

## D. More Results

**Point Cloud Comparisons.** We additionally provide point cloud comparisons of our relative depth model with

other methods, including MoGe, MoGe-2 and PPD. As shown in Fig. 11, our relative depth model demonstrates the strong capability for fine-grained depth estimation.

**Depth Map Comparisons.** We provide more depth map comparisons of our relative depth model with additional baseline methods, as shown in Fig. 12.

**Single-View Novel View Synthesis (NVS) Comparisons.** We present more visual comparisons of single-view NVS



Figure 10. **More RGB images in Synth4K.** Each row from top to bottom shows RGB images from Synth4K’s five games: *CyberPunk 2077*, *Marvel’s Spider-Man 2*, *Miles Morales*, *Dead Island*, and *Watch Dogs*.

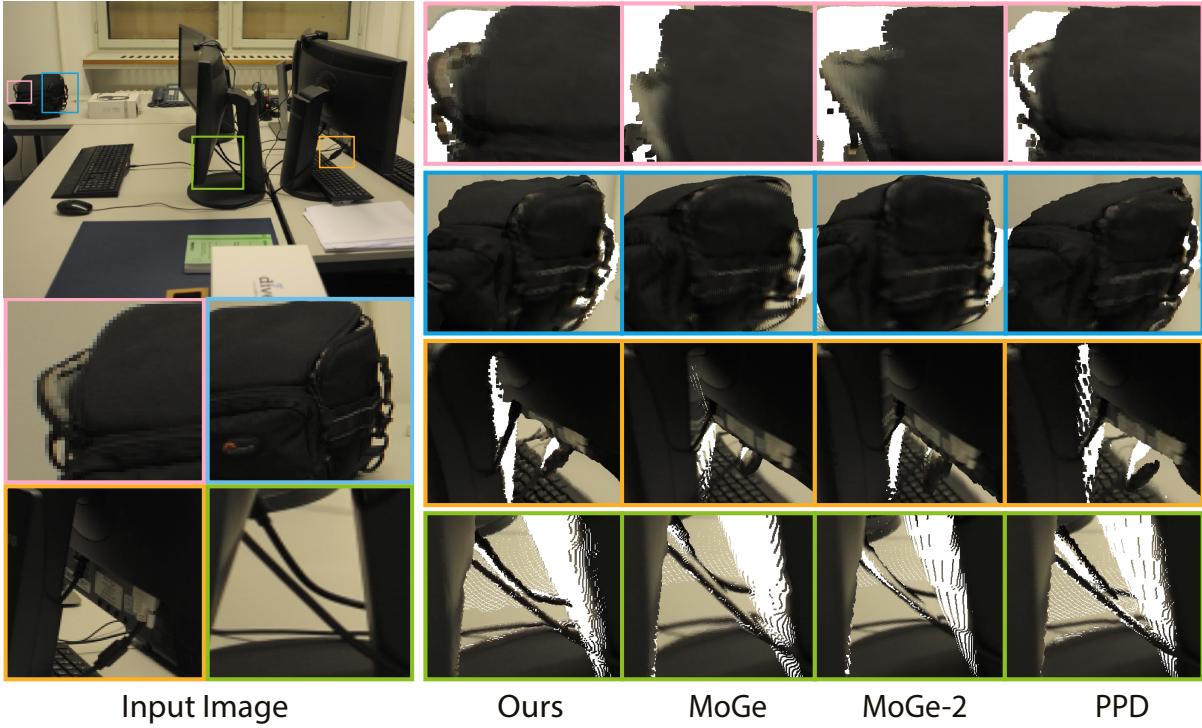
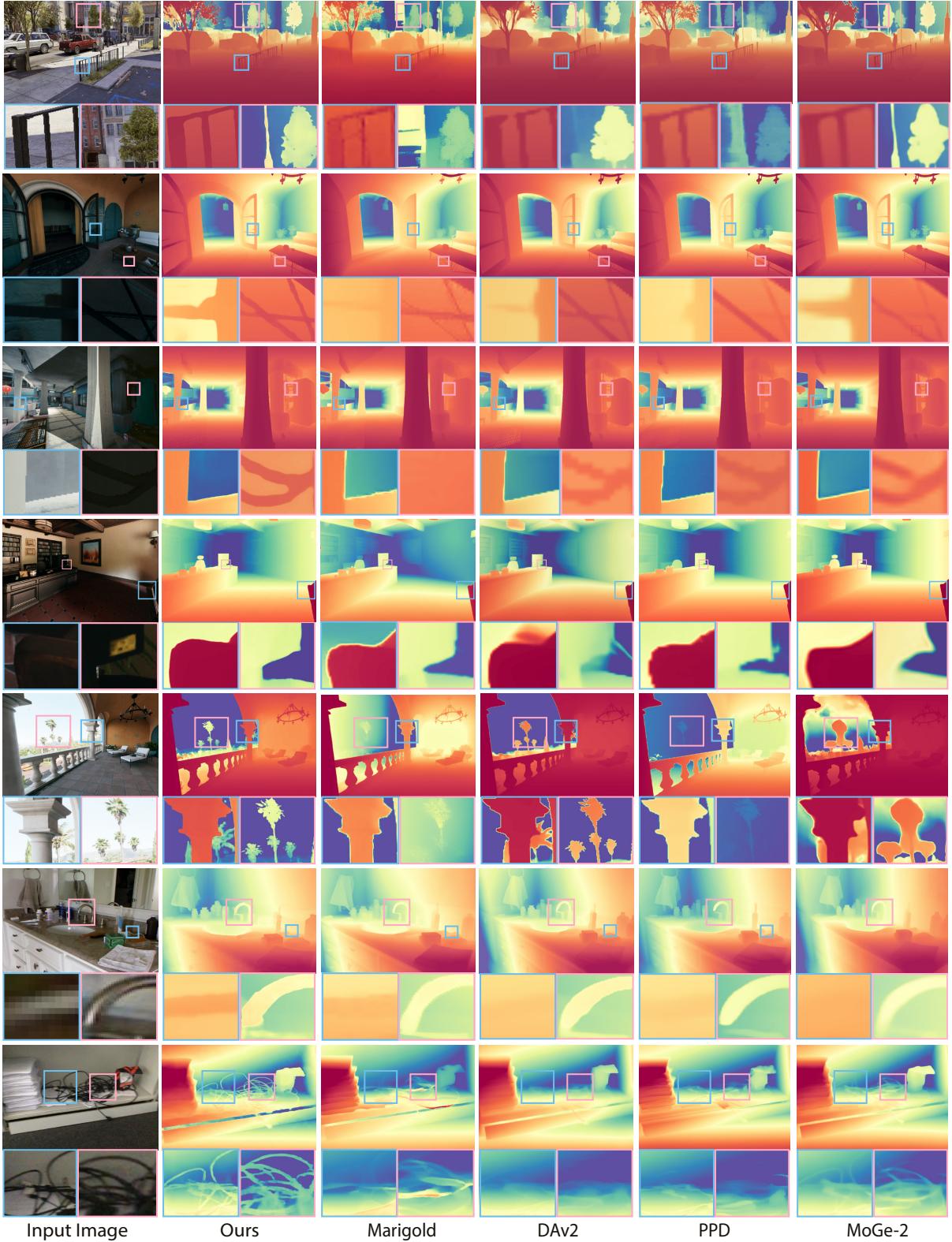


Figure 11. **Point cloud comparisons for relative depth estimation.** Each row from top to bottom shows point clouds predicted by our relative depth model and other SOTA models, including MoGe, MoGe-2 and PPD.



**Figure 12. Depth map comparisons for relative depth estimation.** Each row from top to bottom shows depth maps predicted by our relative depth model and other SOTA models, including Marigold, DepthAnythingV2, PPD and MoGe-2.

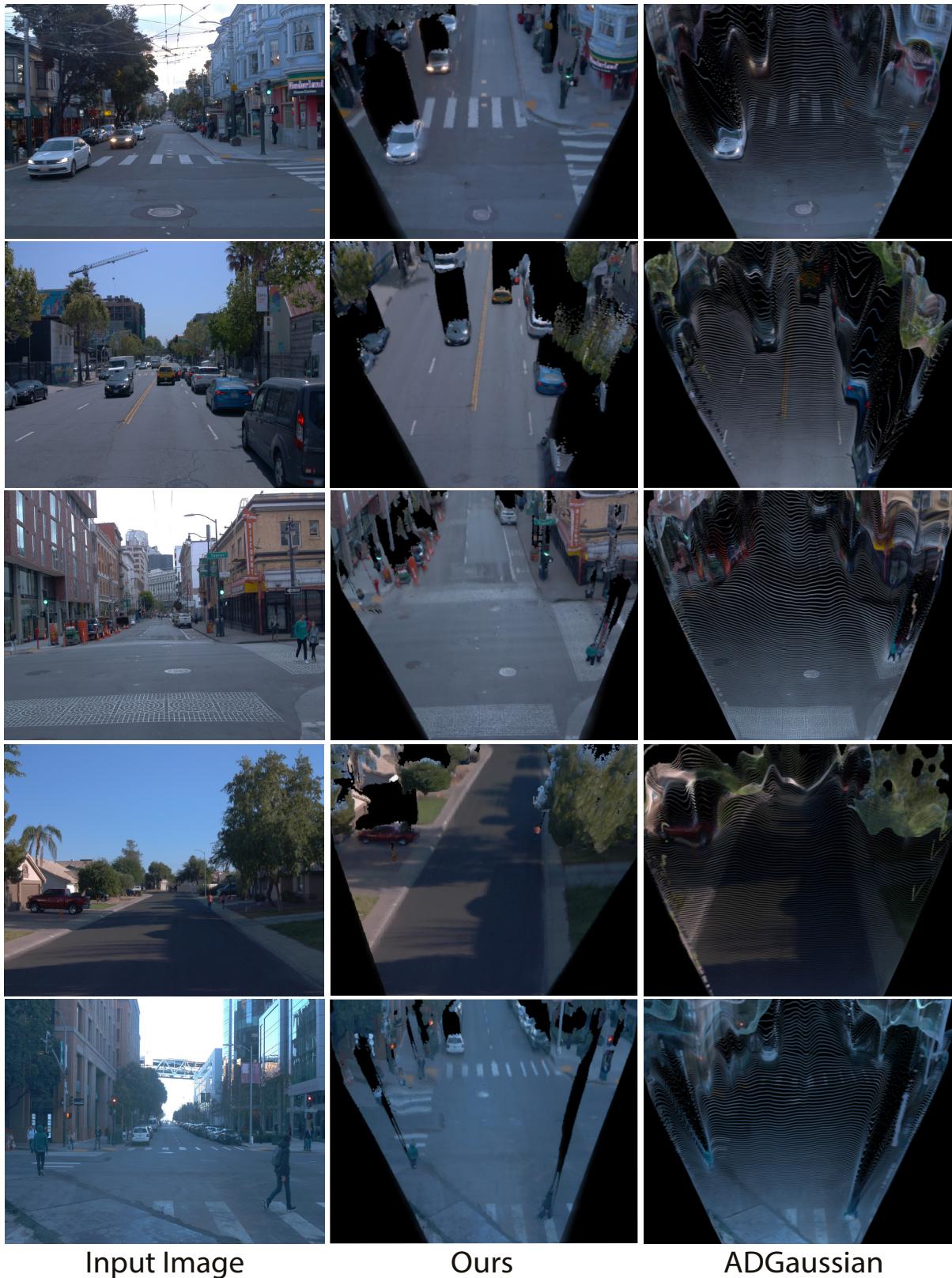


Figure 13. **Single-View Novel View Synthesis (NVS) under large viewpoint shifts.** Each row from top to bottom shows novel view synthesis results from our method and ADGaussian.

results generated by our method and ADGaussian [35] to demonstrate the effectiveness of our proposed depth representation and depth query strategy for this task, as shown in Fig. 13.