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

$$p_i = \frac{w_i}{\sum_i w_i}. \tag{1}$$

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

$$\mathrm{CDF}(k) = \sum_{i=1}^{k} p_i, \tag{2}$$

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

$$q_j = \frac{j + 0.5}{N}, \qquad j = 0, \ldots, N-1, \tag{3}$$

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

$$\mathrm{CDF}(k_j) \geq q_j. \tag{4}$$

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:

$$(x, y) = \big(u{+}0.5{+}\delta_u,\ v{+}0.5{+}\delta_v\big), \quad \delta_u, \delta_v \sim \mathcal{U}(-0.5, 0.5). \tag{5}$$

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 α, 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:

$$d_{norm} = \frac{d_{log} - d_{min}}{d_{max} - d_{min}}, \tag{6}$$

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 $l1$ 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 1×10^{-4}. Supervision combines an $l1$ 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 [1],DepthAnythingV2 [12], MoGe-2 [9], Marigold [8], and PPD [11].

As shown in Tab. 1, 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.

Model	Parameters (M)	Computational Efficiency (s/it)
Ours	15	0.16
DepthPro [1]	29	0.19
DepthAnythingv2 [12]	31	0.03
MoGe-2 [9]	22	0.05
Marigold [8]	-	0.39
PPD [11]	-	1.48

Table 1. **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 504×672 image. We don't report parameters for Marigold and PPD as they are diffusion-based models.

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

$$D_s = \begin{cases} \text{GaussianBlur}(D,\, \sigma = s), & s > 0, \\ D, & s = 0. \end{cases} \tag{7}$$

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

$$\mathcal{L}(D_s) = \left| D_s * \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \right|, \tag{8}$$

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

$$E(x,y) = \max_s \mathcal{L}(D_s)\,(x,y). \tag{9}$$

To suppress extreme outliers, we normalize E using its 98^{th} percentile:

$$\hat{E}(x,y) = \min\left(\frac{E(x,y)}{q_{0.98}(E)},\, 1 \right), \tag{10}$$

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

$$\tilde{E}(x,y) = \hat{E}(x,y)^{1/\tau}. \tag{11}$$

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

$$p(x,y) = \frac{\tilde{E}(x,y)}{\sum_{x,y} \tilde{E}(x,y)}, \tag{12}$$

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. 1 and Fig. 2.

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 [5], MVS-Synth [2], Blendedmvs [13], CREStereo [4], FSD [10], and DynamicReplica [3].

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 504×672 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 504×896 for all 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.

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

Table 2. **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
	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$	$\delta_{0.01}$
Full Model	**82.5**	84.6	84.9	**93.5**	**92.5**	**95.2**	**98.7**	**97.3**	**97.2**	**96.6**
w/o Neural Implicit Fields	82.4	**85.3**	**85.3**	93.4	90.2	94.6	98.3	96.9	97.1	96.1

Table 3. **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	**61.7**	**93.9**	**84.7**	**88.5**	**97.6**
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 4. **Quantitative ablations on different design choices for metric depth estimation.**

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 [7] training split and evaluate it on unseen scenes. Qualitative results are shown in Fig. 5.

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. 2 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. 3. 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 similiar 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 Vs 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. 4, bilinear feature interpolation 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 per-

Figure 1. **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*.

formance 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. 3, 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. 4.

Single-View Novel View Synthesis (NVS) Comparisons. We present more visual comparisons of single-view NVS results generated by our method and ADGaussian [6] to demonstrate the effectiveness of our proposed depth representation and depth query strategy for this task, as shown in Fig. 5.

Figure 2. **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*.

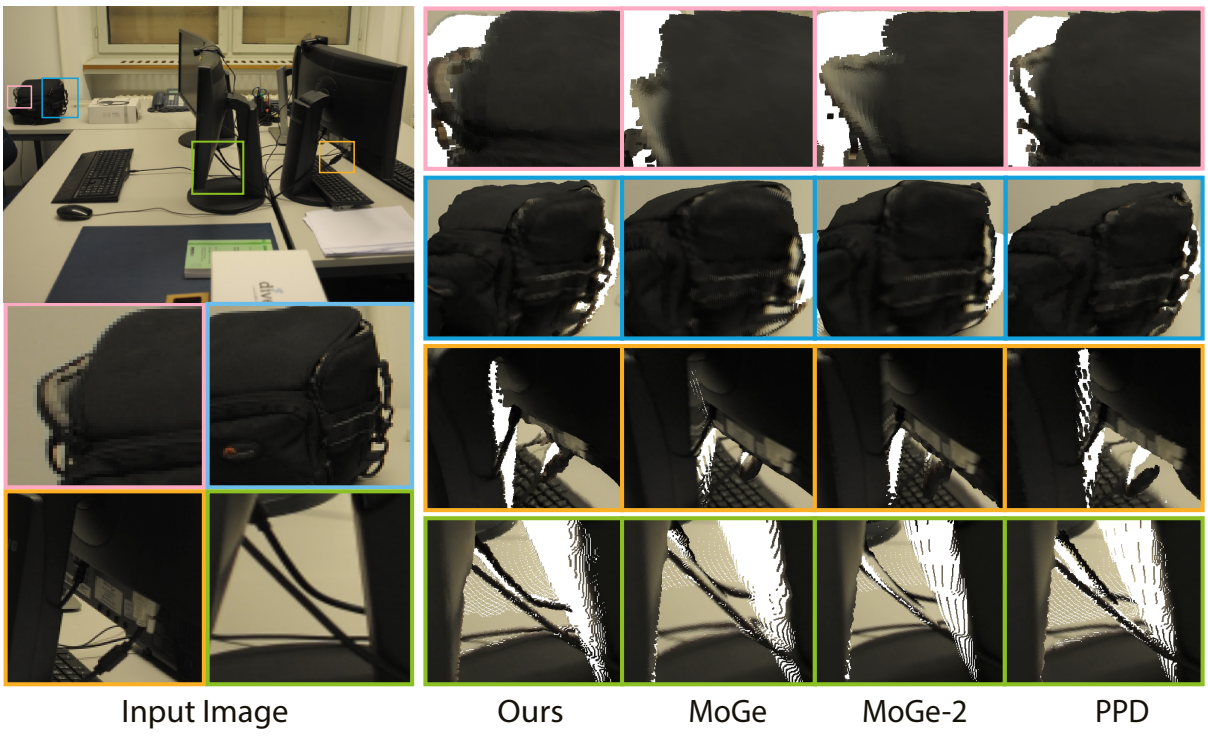

Figure 3. **Point cloud comparisons for relative depth estimation.** Each row from top to bottom shows point clouds predicted by our relative depth model and other SOTA models, including MoGe, MoGe-2 and PPD.

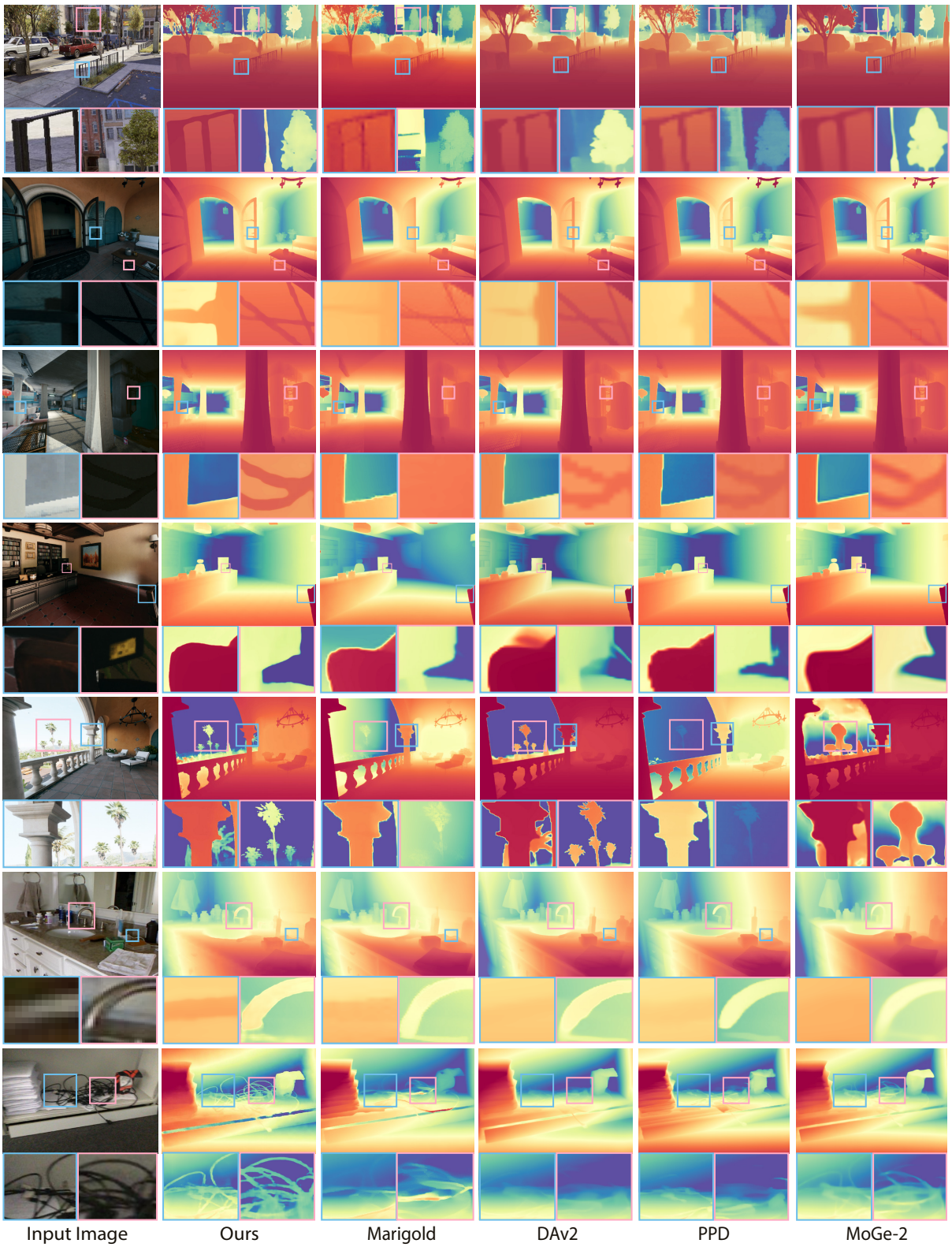

Figure 4. **Depth map comparisons for relative depth estimation.** Each row from top to bottom shows depth maps predicted by our relative depth model and other SOTA models, including Marigold, DepthAnythingV2, PPD and MoGe-2.

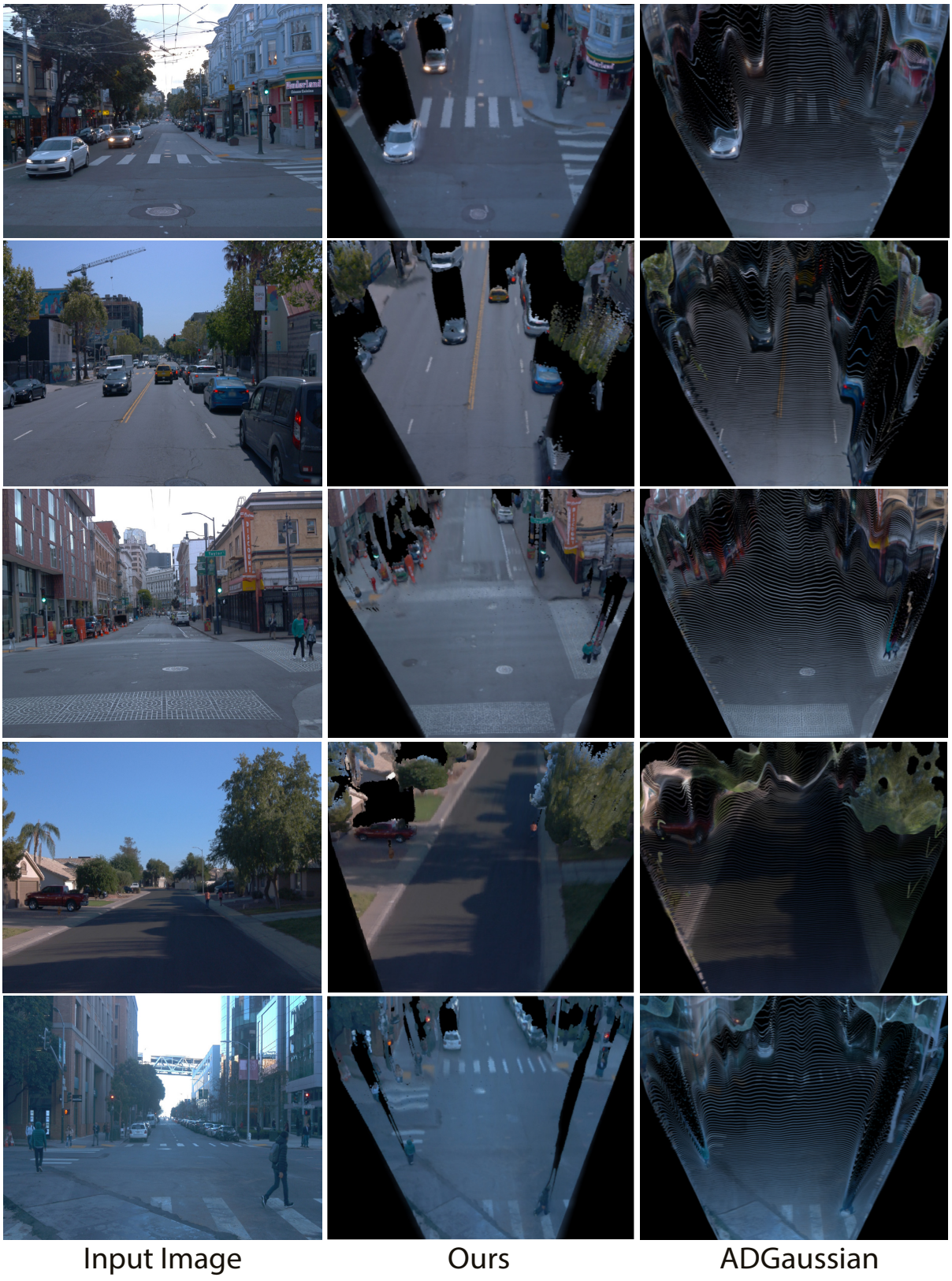

Figure 5. **Single-View Novel View Synthesis (NVS) under large viewpoint shifts.** Each row from top to bottom shows novel view synthesis results from our method and ADGaussian.

References

[1] Aleksei Bochkovskii, AmaÃĢl 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

[2] 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

[3] 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

[4] 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

[5] 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. 2

[6] 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. 4

[7] 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

[8] 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. 2

[9] 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. 2

[10] 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

[11] 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

[12] 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

[13] 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