Appendix for "OMNI-DC: Highly Robust Depth Completion with Multiresolution Depth Integration"

Yiming Zuo, Willow Yang, Zeyu Ma, Jia Deng
Department of Computer Science, Princeton University
{zuoym,willowliuyang,zeyum,jiadeng}@princeton.edu

A. Comparison with Additional Generalizable DC Baselines

The codes are not available for some of the generalizable DC baselines [7, 22], so we are only able to compare against them on NYU and KITTI.

While all termed "generalizable", previous works focus on more restricted settings (TTADC and UniDC on label-free / few-shot domain adaptation; SpAgNet and Depth-Prompting on generalization across sparsity), in contrast to the most challenging zero-shot, sensor-agnostic setting of our paper. As shown in Tab. a, ours (zero-shot) works even better than UniDC tested on the easier 100-shot setting. On NYU, ours even outperform fully-supervised SpAgNet.

Table a. Numbers are copied from original papers when possible. Metric is RMSE. Ours works best under the generalizable settings (*i.e.*, TTA/100-shot/zero-shot) on both datasets and across densities (64Lines & 8Lines on KITTI).

Methods	Setting	NYU-500P	KITTI-64L	KITTI-8L
DPrompting [23]	Fully Supervised	0.105	1.086	1.642
SpAgNet [7]		0.114	0.845	2.691
VPP4DC [1]		0.117	0.099	-
TTADC [21]	Test-Time Adapt.	0.204	-	-
UniDC [22]	100-Shot	0.147	1.224	2.890
Drompting [23]		0.175	1.275	4.587
UniDC [22]	**Zero-Shot**	0.323	4.061	-
VPP4DC [1]		0.247	1.609	-
Ours		**0.111**	**1.191**	**2.058**

B. Ablation Studies on Training Data

We show two things here: 1) Mixing real-world data for training harms performance, both qualitatively and quantitatively. 2) When using the same training datasets, our method still works better than baselines (*i.e.*, OGNIDC [42] and CompletionFormer [41]).

We train OMNI-DC and baselines on either fully synthetic data, or synthetic+NYUv2. As shown in Fig. a, synthetic+real training produces blurry results, as NYU labels from Kinect are blurry. Compared to fully synthetic training, mixing NYU for training results in worse RMSE for zero-shot testing on most (6/8) of the datasets, as shown in Tab. c. Nevertheless, ours is still better than baselines when trained on synthetic+real (RMSE reduced by 29.2% from OGNI-DC and by 36.1% from CFormer on KITTI).

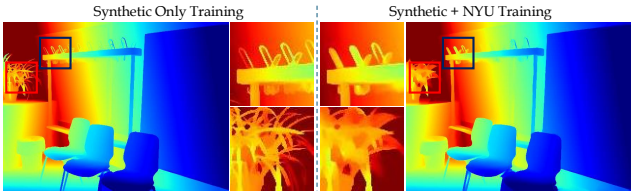

Figure a. Mixing NYU for training produces blurry depth maps on iBims [14].

C. Results on Radar Depth Completion

To show that OMNI-DC can generalize beyond the sparse depth patterns that it was trained on, we evaluate on the Radar-Camera fusion benchmark, ZJU-4DRadarCam [15]. As shown in Tab. b, ours outperforms all zero-shot baselines. While baselines such as G2-MD also claim to be generalizable, they perform much worse. As shown in Fig. b, while our metrics slightly fall behind RadarCam-Depth (which is trained in-domain and not zero-shot), our depth map is much sharper. Sharpness is crucial for novel view synthesis applications to avoid boundary artifacts.

Table b. We follow [15] and test with three ranges. Numbers in gray are trained on ZJU-4DRadarCam and are not zero-shot; others are zero-shot. Ours outperforms all other zero-shot methods, though it falls behind methods trained in-domain.

GT-ranges	50m		70m		80m	
	RMSE↓	iRMSE↓	RMSE↓	iRMSE↓	RMSE↓	iRMSE↓
DORN [17]	4129.7	31.853	4625.2	31.877	4760.0	31.879
Singh *et al.* [29]	3704.6	35.342	4137.1	35.166	4309.3	35.133
RadarCam [15]	2817.4	22.936	3117.7	22.853	3229.0	22.838
DA-v2 [40]	5466.5	47.446	6261.3	47.118	6566.9	47.053
OGNI-DC [42]	7612.7	29107.2	8151.2	28800.5	8356.9	28739.0
G2-MD [34]	7237.2	61.285	7980.3	60.803	8232.3	60.717
Ours	**5256.8**	**41.477**	**5984.1**	**41.253**	**6249.1**	**41.207**

Table c. Ablation studies on the effect of mixing real-world dataset for training. NYU consists of 1/6 of all data. All models are trained with 1/10 of the full training steps, due to resource constraints. The metric is RMSE, and the sparse depth has 0.7% density except for NYU, VOID, and KITTI. Mixing real training data has a negative effect on most of the datasets, especially obvious outdoor. Ours works better than OGNI-DC and CFormer under both training settings.

	In-Domain	Zero-Shot, Indoors					Zero-Shot, Outdoors		
Training	NYU-500P	iBims	ETH3D(In)	DIODE(In)	ARKitScenes	VOID-1500P	ETH3D(Out)	DIODE(Out)	KITTI-64L
OMNI-DC, Synthetic Only (**Ours**)	0.119	**0.156**	**0.118**	**0.056**	0.023	**0.565**	**0.322**	**2.307**	**1.279**
OMNI-DC, Synthetic + NYU	**0.110**	**0.156**	0.119	0.058	**0.022**	0.567	0.324	2.337	1.309
OGNI-DC [42], Synthetic Only	0.125	0.164	0.124	0.063	0.024	0.573	0.333	2.332	1.846
OGNI-DC [42], Synthetic + NYU	0.120	0.166	0.127	0.064	0.023	0.595	0.337	2.411	1.850
CFormer [41], Synthetic Only	0.130	0.176	0.148	0.064	0.030	0.595	0.359	2.338	2.037
CFormer [41], Synthetic + NYU	0.128	0.173	0.148	0.066	0.025	0.627	0.388	2.382	2.047

Table d. Our method is robust under challenging imaging conditions (*e.g.*, nighttime and different weathers).

Datasets	Carla-Night-DC [38]				DS-Sunny [39]		DS-Rainy [39]		DS-Foggy [39]		DS-Cloudy [39]	
	RMSE↓	MAE↓	iRMSE↓	iMAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓
LDCNet [38]	7.214	2.014	0.0546	0.0156	-	-	-	-	-	-	-	-
DA-v2 [40]	104.878	68.242	0.1560	0.0976	7.544	2.941	7.567	3.805	7.868	2.927	8.252	2.964
OGNI-DC [42]	13.576	5.469	0.2191	0.0738	3.774	1.494	5.730	2.384	3.756	1.654	3.903	1.499
G2-MD [34]	10.488	3.291	0.0930	0.0246	3.013	0.875	2.809	0.982	3.130	1.149	3.053	0.872
Ours	**10.068**	**2.523**	**0.0413**	**0.0105**	**2.765**	**0.741**	**2.645**	**0.844**	**2.744**	**0.909**	**2.735**	**0.714**

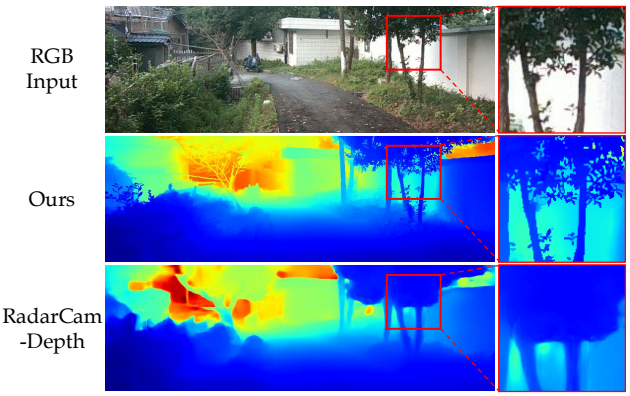

Figure b. Results on the Radar depth completion task. Our depth maps are much sharper than RadarCam-Depth [15].

D. Robustness to Night Time and Bad Weather

As shown in Tab. d, our method is robust.

Carla-Night-DC [38] contains night-time driving scenes. LDCNet [38] is trained on Carla-Night-DC and other methods are tested zero-shot. Our method works the best, even outperforming LDCNet on iRMSE and iMAE despite never being trained specifically on night scenes.

The DrivingStereo (DS) [39] dataset consists of real images from driving scenes captured at different weathers. We randomly sample 500 points from GT as sparse depth. Our method consistently outperforms baselines under all weather conditions, and is more robust (OGNI-DC's RMSE ↑ 52% under "Rainy" than "Sunny", while ours' RMSE ↓ 4%.)

E. Details on Novel View Synthesis

In the main paper, we have shown a practical downstream application of OMNI-DC on novel view synthesis. Training neural rendering frameworks such as NeRF [19] or 3DGS [12] on sparse input views is a challenging task, and introducing geometric priors such as depth as a regularization has been shown helpful in previous works [6, 8]. We follow the recent work DN-Splatter [31], and use a depth loss to train 3DGS. The loss can be written as:

$$\mathcal{L} = \mathcal{L}_{\hat{C}} + 0.2 \cdot \mathcal{L}_{\hat{D}}, \tag{1}$$

where $\mathcal{L}_{\hat{C}}$ is the original photometric loss in 3DGS [12], and $\mathcal{L}_{\hat{D}}$ is the edge-aware depth loss proposed in [31].

We evaluate on the ETH3D [28] dataset with 13 scenes, each containing 14-76 images. The scales of the scenes are large, creating a challenging sparse view setting. We compare against the vanilla 3DGS with no depth supervision, as well as supervising with the depth map obtained from the monocular depth model ZoeDepth [3], and the depth completion model G2-MD [34]. For ZoeDepth, we align the scale and shift against the COLMAP sparse depth, following DN-Splatter [31]. For G2-MD and our method, we run depth completion on the COLMAP sparse depth. In addition to the results presented in the paper, we also compare against the state-of-the-art multi-view stereo (MVS) method, MVSFormer++ [5].

We randomly split $1/8$ of the view as test views and use the rest for training. The training follows the [31] schedule for 30K steps. We have reported the image quality statistics PSNR, SSIM, and LPIPS, as well as the RMSE between the rendered depth and the ground-truth depth on test views.

Table e. The novel view synthesis metrics and the depth accuracy averaged on the 13 scenes from ETH3D.

Methods	3DGS	Zoe-Depth	G2-MD	MVS-Former++	Ours
PSNR ↑	15.64	18.96	19.36	20.02	**20.38**
SSIM ↑	0.557	0.573	0.641	0.644	**0.660**
LPIPS ↓	0.418	0.324	0.273	0.254	**0.229**
RMSE (Depth) ↓	3.857	2.163	1.904	1.847	**0.838**

As shown in Tab. e, OMNI-DC outperforms all methods in terms of both rendering and geometry reconstruction quality.

More visualizations are shown in Fig. c. The 3DGS regularized with our depth maps produces much fewer floater artifacts compared to baselines. This shows that users can directly use our OMNI-DC to improve the 3DGS quality, without any retraining for the depth model.

F. Implementation Details

F.1. Model Architecture and Loss Functions

We use the CompletionFormer [41] as the backbone. CompletionFormer is a U-Net-like [25] architecture with a feature pyramid. We extract the depth gradients by using the $1/4$ resolution feature map with a series of ResNet [9] blocks and $\mathrm{MaxPool2D}$ layers, to obtain the depth gradients at the $1/4$, $1/8$, and $1/16$ resolution.

From the full-resolution feature map, we extract the parameters for the DySPN [16] (propagation weights and confidence) and scale parameters for computing the Laplacian loss. Specifically, since the scale parameter b must be positive, we parameterize it as $b = \exp(\gamma)$ following [36], and predict γ from a Conv layer. We clamp the minimum value of γ to -2.0 to stabilize training.

To better deal with the noise in the input depth, we follow OGNI-DC [42] and use a $\mathrm{sigmoid}$ layer to predict a confidence map for the input sparse depth. Denote the confidence map as $\hat{\mathbf{C}} \in (0,1)^{H \times W}$, the sparse depth energy term is re-weighted as (see Eqn.3 in the main paper):

$$\mathcal{E}_O = \sum_{i,j}^{W,H} \mathbf{M}_{i,j} \cdot \mathbf{C}_{i,j} \cdot (\mathbf{D}_{i,j} - \mathbf{O}_{i,j})^2 \tag{2}$$

When $C_{i,j} \to 0$, the contribution of the corresponding sparse depth point becomes zero, providing a data-driven mechanism for the network to ignore the noisy depths. Unlike OGNI-DC which trains the confidence map through the depth loss, we record the noisy pixels when generating the virtual sparse pattern and use an axillary binary cross-entropy loss to directly supervise the confidence map.

The gradient-matching loss is implemented following MegaDepth [24] and MiDaS [24]:

$$\mathcal{L}_{\text{gm}} = \frac{1}{HW} \sum_{k=1}^{4} \sum_{i,j}^{W,H} \left(\left| \nabla_x R^k_{i,j} \right| + \left| \nabla_y R^k_{i,j} \right| \right), \tag{3}$$

Where $R^1 = \hat{\mathbf{D}} - \mathbf{D}^{\text{gt}}$. Similarly, R^k is the depth difference at the k^{th} resolution.

F.2. Training Details

The model is trained with an Adam [13] optimizer with an initial learning rate of $1e-3$, for a total of 72 epochs. The learning rate decays by half at the 36$^{\text{th}}$, 48$^{\text{th}}$, 56$^{\text{th}}$, and 64$^{\text{th}}$ epochs, following [41].

Since the five training datasets are vastly different in size, we uniformly sample 25K images from each dataset to balance their contributions in each epoch. We also normalize the median depth values of all training samples to 1.0 to balance the loss among different types of scenes.

We sample the random samples, SfM keypoints, and LiDAR points with a ratio of 2:1:1. This ratio empirically yeilds good performance, but the performance of our model is not sensitive to it. Random point densities are sampled in the range $0.03\% \sim 0.65\%$ (*i.e.*, $100 \sim 2000$ points). The SfM points are sampled at the SIFT [18] keypoints. For the random and SfM points, we also inject $0\% \sim 5\%$ noisy depths by random sampling between the 5$^{\text{th}}$ and 95$^{\text{th}}$ percentile interval of the image depth range. When generating the LiDAR keypoints, we randomize the number of lines, the center of the LiDAR, and the camera intrinsics. We additionally synthesize the boundary error caused by the baseline between the camera and the LiDAR. Specifically, we random sample a virtual viewpoint for the LiDAR., and project the depth to the virtual view. This leaves holes in the projected depth map, so we use the heuristic-based inpainting used in LRRU [35] to fill those holes. We finally sample the LiDAR points from the virtual view, and project it back to the original view.

G. Limitations

1) Like other depth estimation models, our method faces challenges when predicting depth for transparent surfaces (*e.g.*, glasses), reflective surfaces, or the sky. In Fig. d we show a few failure cases. **2)** The backbone of our method takes 4 channels (RGB-D) input, which makes it hard to benefit from the pre-trained models designed for RGB images, such as DINO-v2 [20]. One possible direction is removing the depth channel from the feature extractor. **3)** Our model currently cannot deal with the case with no sparse depth inputs (*i.e.*, monocular depth estimation). Having the model's performance degrade more smoothly when the input depths become sparser is a future direction. **4)** The current model cannot handle certain types of sparse depth patterns very well, such as the radar inputs discussed

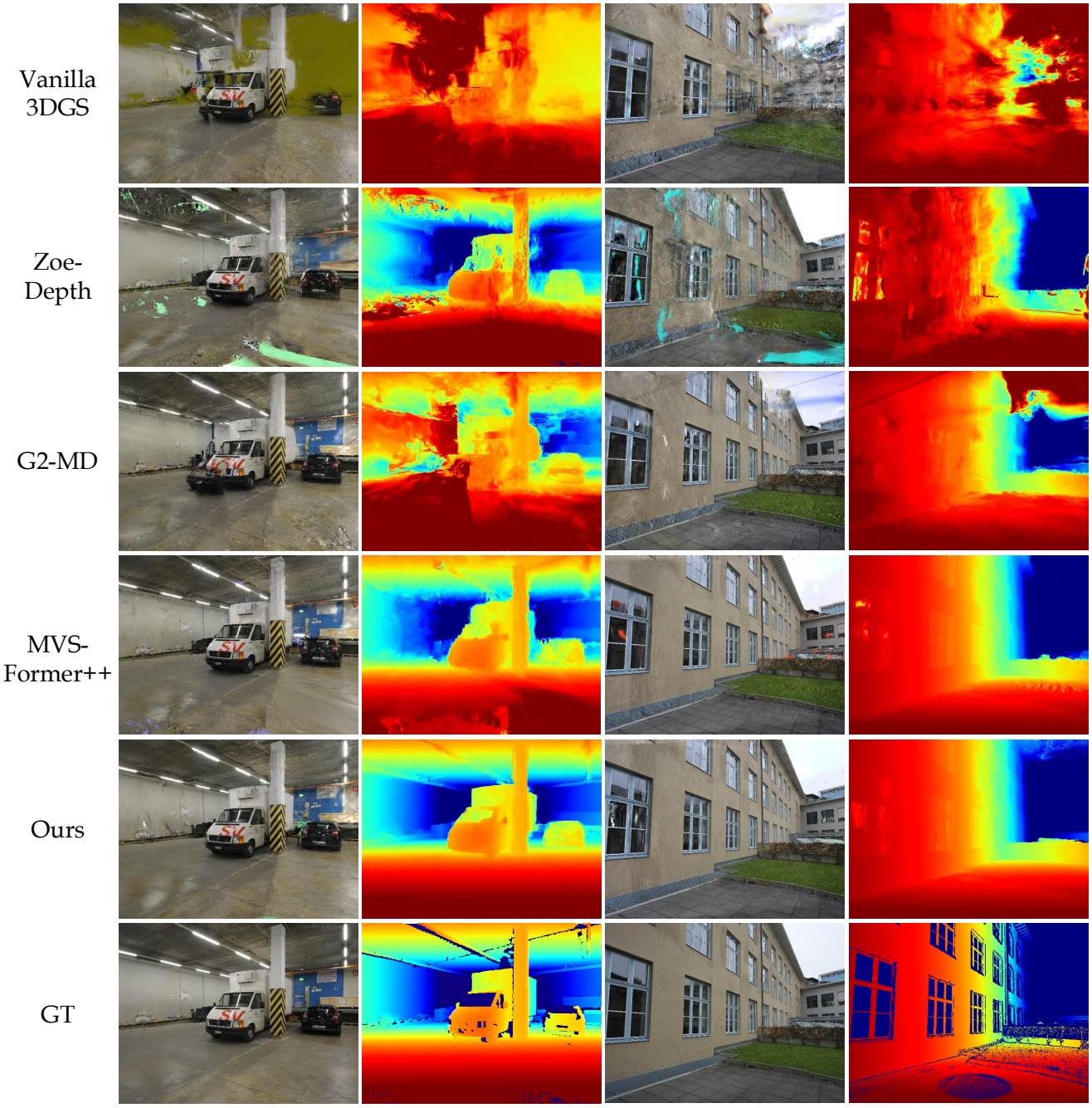

Figure c. Visualization of the rendered images and rendered depth maps against ground-truth on test views of the ETH3D dataset. The vanilla 3DGS is trained with only the photometric loss, and all other rows are trained with a depth loss against the predicted depth maps of the corresponding models. Our model generates significantly higher quality images and geometry (depth maps).

in Sec. C, and large holes that may appear in object removal/inpainting applications. Expanding the sparse depth synthesis pipeline to cover these during training is a promising direction.

H. More Results on the NYUv2 Dataset

We show results on more densities in Tab. f. We exclude all the in-domain DC baselines trained on the NYU training set from the ranking. Our method works better than all zero-shot baselines on the 500, 200, 100, and 50 densities. On the original setting of NYUv2 (NYU-500), our method has a close performance to the best model trained on NYU (REL=0.014 vs 0.011 for DFU [37]). On the extremely sparse case (NYU-5), our method works better than OGNI-DC [42] and G2-MD [34], although worse than the monocular depth methods such as Depth Pro [4].

Table f. Results on the NYUv2 dataset with 5-500 random samples. The numbers in gray are trained on NYU with 500 points, and we exclude them from the ranking. On relatively dense inputs, our method works the best among all the methods tested zero-shot, and is very close to the best model trained on NYU (REL=0.014 vs 0.011 for DFU [37] on NYU-500). On NYU-5, our method works better than all DC baselines (RMSE=0.536 vs 0.633 for OGNI-DC [42]).

	Methods	NYU-500		NYU-200		NYU-100		NYU-50		NYU-5	
		RMSE	REL	RMSE	REL	RMSE	REL	RMSE	REL	RMSE	REL
Trained on NYU	CFormer [41]	0.090	0.012	0.141	0.021	0.429	0.092	0.707	0.181	1.141	0.307
	DFU [37]	0.091	0.011	-	-	-	-	-	-	-	-
	BP-Net [30]	0.089	0.012	0.132	0.021	0.414	0.090	0.609	0.157	0.869	0.294
	DPromting [23]	0.105	0.015	0.144	0.023	0.178	0.031	0.213	0.043	0.380	0.095
	OGNI-DC [42]	0.089	0.012	0.124	0.018	0.157	0.025	0.207	0.038	0.633	0.171
Zero-shot	Depth Pro [4]	0.266	0.062	0.266	0.062	0.266	0.062	0.266	0.062	0.266	0.062
	DA-v2 [40]	0.309	0.061	0.309	0.061	0.314	0.062	0.330	0.063	0.814	0.136
	Marigold [11]	0.426	0.115	0.428	0.116	0.431	0.117	0.436	0.118	0.545	0.150
	G2-MD [34]	0.122	0.017	0.169	0.027	0.222	0.038	0.286	0.056	0.744	0.207
	Ours	0.111	0.014	0.147	0.021	0.180	0.029	0.225	0.041	0.536	0.142

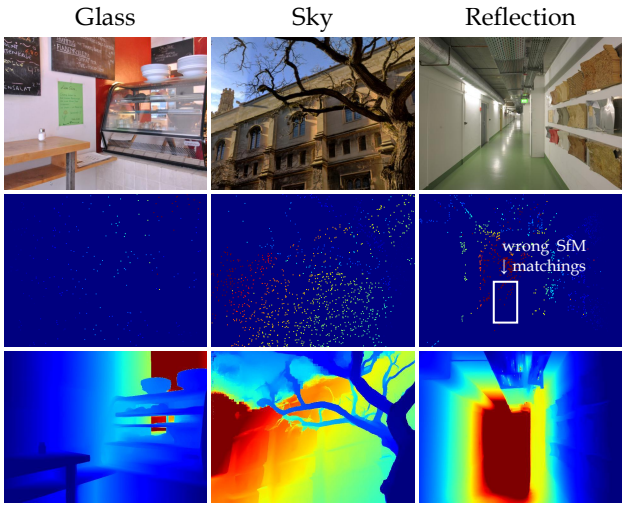

Figure d. Failure cases of OMNI-DC. Our model makes erroneous predictions when the scene contains glasses or reflective surfaces, as the depth sensor or multiview matching may fail. The sky cannot be naturally represented in the linear depth space.

I. Visualizations of Point Cloud

We visualize the 3D reconstruction quality of our predicted depth map by projecting the depth map into 3D using the ground-truth camera intrinsics. We also compared against the few strongest baselines, *i.e.*, DepthAnythingv2 [40], OGNI-DC [42], and G2-MonoDepth [34]. As shown in Fig. e, our method achieves better results in both global structures (orientation of the walls) and local details (cars).

J. Details on Evaluation Datasets

We list the details of the datasets we use below. Samples from the datasets can be found in Figs. g to i.

iBims [14] consists of 100 indoor scenes captured with a laser scanner. The original images are at 480×640 resolution.

ARKitScenes [2] is a large scale dataset consisting of more than 450K frames of scans of 5K indoor scenes. The validation split contains about 3.5K images in the landscape orientation, from which we randomly pick 800 images as our test set. The original high-res laser-scan images are at resolution 1440×1920, from which we resize to 480×640.

ETH3D [28]'s test set contains 13 scenes total with 454 images, with ground-truth captured using a laser scanner. The original images are at 4032×6048 resolution, from which we downsample at approximately a factor of 8 to 480×640. We pick the "office" and the "courtyard" scene as the validation set, and further split the rest 11 scenes into indoors (6 scenes, 193 images) and outdoors (5 scenes, 197 images). For the real SfM patterns, we project the visible keypoints from the COLMAP [27] reconstruction for each scene into 2D to construct the sparse depth map.

DIODE [33]'s validation split contains 3 indoor scenes and 3 outdoor scenes, with 325 and 446 images in total respectively. The ground truth is captured with a FARO laser scanner. We find that the original depth measurements at occlusion boundaries are very noisy. Therefore, we filter out the pixel whose depth is different from its neighboring pixels by more than 5% (indoor) and 15% (outdoor). This effectively removes the noise while preserving most of the useful information. Images are resized to 480×640.

KITTI [32]'s validation set contains 1000 images from 5 scenes in total. We subsample the original 64-line LiDAR by clustering the elevation angles of the LiDAR points to construct the virtual 16-line and 8-line input following [10]. We crop the top 96 pixels containing only sky regions, resulting in an image resolution of 256×1216.

K. Test-Time Scaling Up to Higher-Resolution Images

Most of the experiments in this paper are conducted under the resolution of 480×640. However, modern cameras can

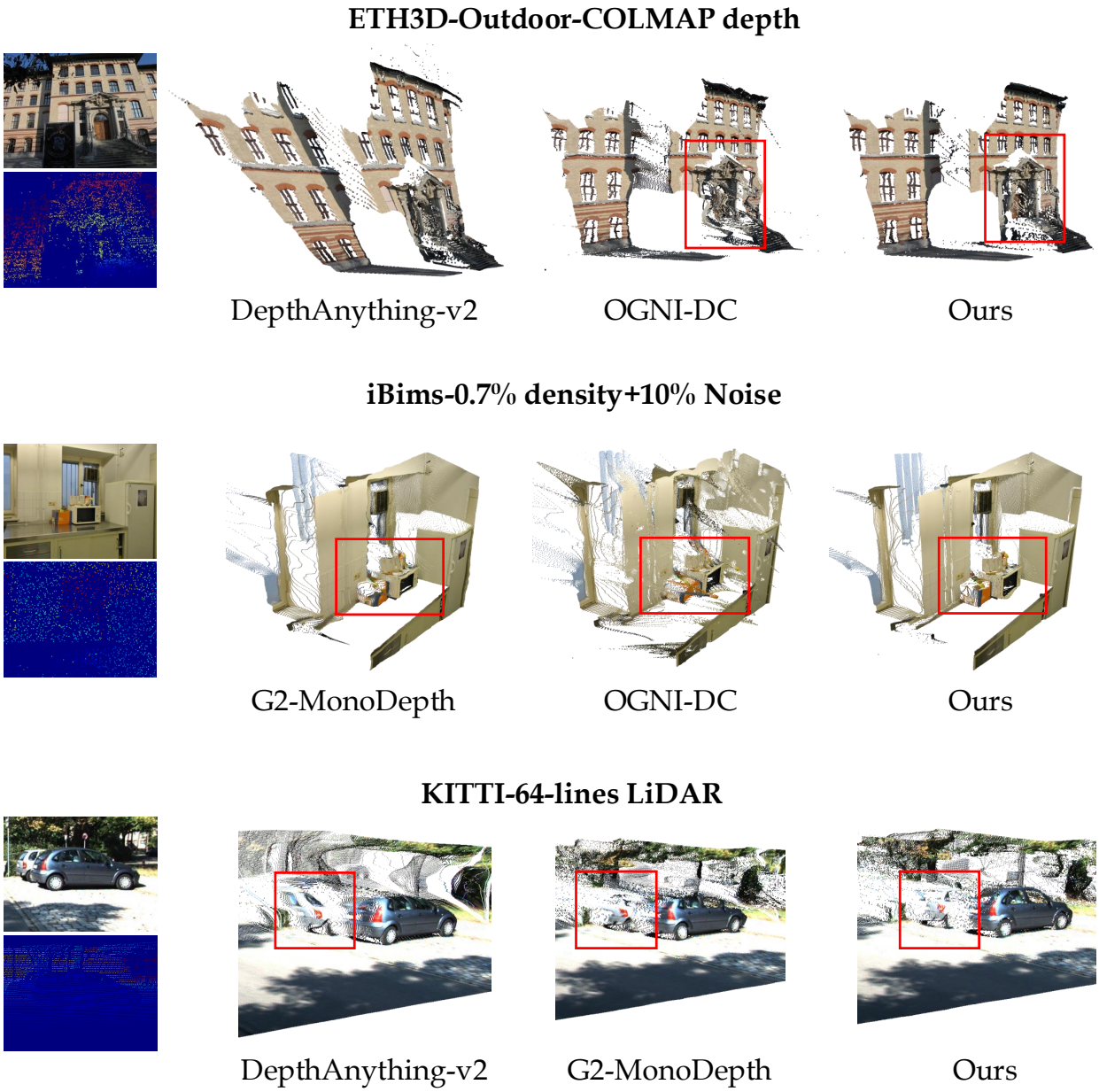

Figure e. The qualitative comparison of the 3D structures between our method and the best-performing baselines. On the outdoor scene from ETH3D, DA-v2 [40] has trouble capturing the global structure, while OGNI-DC's reconstruction has distorted local details. On the noisy sparse depth map on iBims, the OGNI-DC's prediction is greatly distorted by the outliers, and our method is robust to noise. On KITTI, our method is able to reconstruct the high-quality 3D structure of the white car.

often capture images at a higher resolution, which captures more details. Therefore, it is desirable that our DC model can work under higher resolutions.

We feed OMNI-DC with high-resolution images at test time. As shown in Tab. g, the inference time is 2.1× and 3.6× longer when tested on images with 2× and 2.7× resolution, respectively, a lower rate compared to the increase in pixel count. The memory consumption is 11.1GB when tested under the resolution of 1280×1706, which can be held on a 12GB GPU such as an RTX 4070.

Qualitative results are shown in Fig. f. While OMNI-DC is trained on a low resolution (480×640), it can generalize to higher resolution images at test time, producing higher quality depth maps.

The results show that OMNI-DC has a strong capability of scaling up to higher-resolution images at test time.

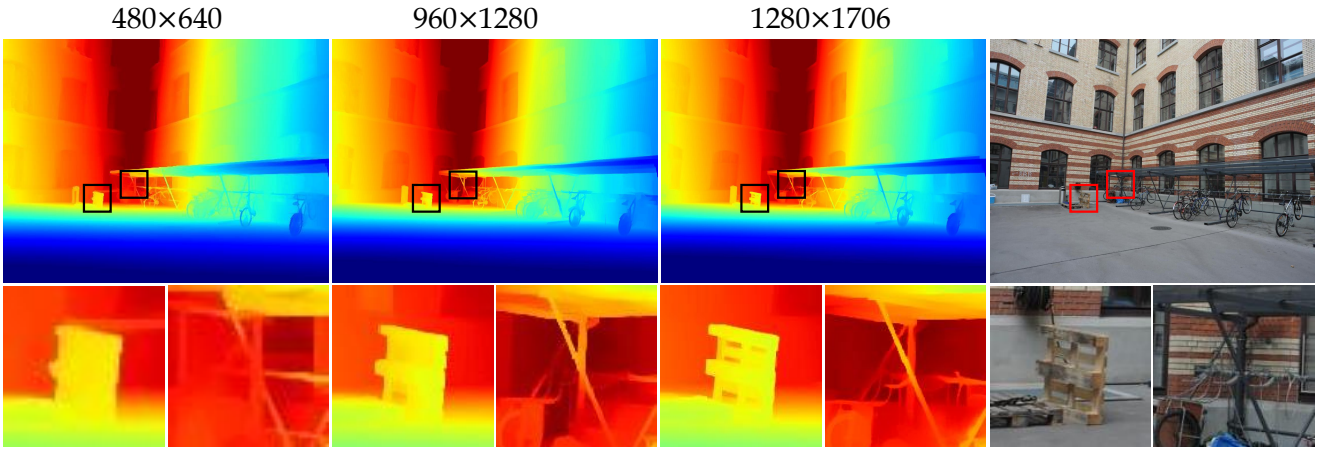

Figure f. More details are captured when running inference with higher resolution images at test time. All sparse depths are sampled under the 0.7% density.

Table g. Speed ane memory consumption on higher resolutions. Numbers benchmarked on a 3090 GPU.

Resolution	480×640	960×1280	1280×1706
Inference Time (ms)	235	495	839
Memory (GB)	4.6	7.9	11.1

L. Guaranteed Scale Equivariance

Scale equivariance means the scale of the output depth respects the scale of the input depth. For example, when the input is given in the unit of millimeters (mm), the output should also be in millimeters. This is a desired property, as it makes the system simple to use. For example, if a DC model is not scale-equivariant, the user will have to convert it to metric space before feeding it into the DC model, which requires estimating the arbitrary scale factor from their COLMAP reconstruction and could be impossible.

Assume F to be a DC model taking the RGB image $\mathbf{I}$ and the sparse depth map $\mathbf{O}$ as input, and outputs a dense depth map $\hat{\mathbf{D}}$, *i.e.*,

$$\hat{\mathbf{D}} = F(\mathbf{I}, \mathbf{O}). \tag{4}$$

We formally define the equivariance property as follows:

$$F(\mathbf{I}, \beta \cdot \mathbf{O}) = \beta \cdot F(\mathbf{I}, \mathbf{O}), \forall \beta \in \mathbb{R}_+, \tag{5}$$

where β is an arbitrary scale factor. For example, $\beta = 1000$ when converting depth from meters (m) into millimeters (mm).

We first theoretically prove that OMNI-DC is guaranteed to be scale equivariant, and then confirm it by empirical results.

L.1. Theoretical Proof

We first show that the input to the neural network is *invariant* to the scale of the input depth. Recall that we normalize the input depth values to the neural network by its median:

$$\hat{\mathbf{G}} = F(\mathbf{I}, \tilde{\mathbf{O}}; \theta), \tilde{\mathbf{O}} = \log(\mathbf{O}) - \log(\text{median}(\mathbf{O})). \tag{6}$$

It is easy to see that $\tilde{\mathbf{O}}$ is invariant to the input scale, *i.e.*,

$$\begin{aligned} &\tilde{\mathbf{O}}(\beta \cdot \mathbf{O}) = \log(\beta \cdot \mathbf{O}) - \log(\text{median}(\beta \cdot \mathbf{O})) \\ &= \log(\beta) + \log(\mathbf{O}) - \log(\beta) - \log(\text{median}(\mathbf{O})) \\ &= \tilde{\mathbf{O}}(\mathbf{O}), \forall \beta \in \mathbb{R}_+. \end{aligned} \tag{7}$$

Correspondingly, the output of the neural network, $\hat{\mathbf{G}}$, is also invariant to the input scale, because all its input is scale-invariant:

$$\hat{\mathbf{G}}(\mathbf{I}, \beta \cdot \mathbf{O}) = \hat{\mathbf{G}}(\mathbf{I}, \mathbf{O}), \forall \beta \in \mathbb{R}_+. \tag{8}$$

We therefore omit the input of $\hat{\mathbf{G}}$ and treat it as a constant in the following deductions.

Note that the depth integration is done in the log-depth space, and recall the energy terms are:

$$\hat{\mathbf{D}}^{\log} = \arg\min_{\mathbf{D}^{\log}} \left(\alpha \cdot \mathcal{E}_O(\mathbf{D}^{\log}, \mathbf{O}, \mathbf{M}) + \mathcal{E}_G(\mathbf{D}^{\log}, \hat{\mathbf{G}}) \right), \tag{9}$$

where

$$\begin{aligned} \mathcal{E}_O &:= \sum_{i,j}^{W,H} \mathbf{M}_{i,j} \cdot (\mathbf{D}^{\log}_{i,j} - \log(\mathbf{O}_{i,j}))^2, \\ \mathcal{E}_G &:= \sum_{r=1}^{R} \sum_{i,j}^{W,H} \left(\mathbf{G}^x_{i,j} - \hat{\mathbf{G}}^x_{i,j} \right)^2 + \left(\mathbf{G}^y_{i,j} - \hat{\mathbf{G}}^y_{i,j} \right)^2, \end{aligned} \tag{10}$$

with $\mathbf{G}^{r,x}_{i,j} := \mathbf{D}^r_{i,j} - \mathbf{D}^r_{i-1,j}$; $\mathbf{G}^{r,y}_{i,j} := \mathbf{D}^r_{i,j} - \mathbf{D}^r_{i,j-1}$ being the analytical gradients at the resolution r.

We write $\hat{\mathbf{D}}^{\log}$ as a function of $\hat{\mathbf{G}}$, $\mathbf{O}$, and $\mathbf{M}$, i.e., $\hat{\mathbf{D}}^{\log}(\hat{\mathbf{G}}, \mathbf{O}, \mathbf{M})$. Given the above definition, we have the lemma below:

Lemma 1 *If* $\hat{\mathbf{D}}^{\log}(\hat{\mathbf{G}}, \mathbf{O}, \mathbf{M})$ *is the optimal solution to Eq. (9), then* $\log \beta + \hat{\mathbf{D}}^{\log}(\hat{\mathbf{G}}, \mathbf{O}, \mathbf{M})$ *is the optimal solution if we multiply* $\mathbf{O}$ *by* β*,* i.e.*,* $\hat{\mathbf{D}}^{\log}(\hat{\mathbf{G}}, \beta \cdot \mathbf{O}, \mathbf{M}) = \log \beta + \hat{\mathbf{D}}^{\log}(\hat{\mathbf{G}}, \mathbf{O}, \mathbf{M}), \forall \beta \in \mathbb{R}_+$.

This can be seen from the linearity of Eq. (10). Plugging $\log \beta + \mathbf{D}^{\log}$ and $\beta \cdot \mathbf{O}$ into Eq. (10) gives the exact same energy as $\mathbf{D}^{\log}$ and $\mathbf{O}$.

Given Lemma 1, we finally have

$$\begin{aligned} \hat{\mathbf{D}}(\hat{\mathbf{G}}, \beta \cdot \mathbf{O}, \mathbf{M}) &= \exp\left(\hat{\mathbf{D}}^{\log}(\hat{\mathbf{G}}, \beta \cdot \mathbf{O}, \mathbf{M})\right) \\ &= \exp\left(\log \beta + \hat{\mathbf{D}}^{\log}(\hat{\mathbf{G}}, \mathbf{O}, \mathbf{M})\right) \\ &= \beta \cdot \hat{\mathbf{D}}(\hat{\mathbf{G}}, \mathbf{O}, \mathbf{M}), \forall \beta \in \mathbb{R}_+. \square \end{aligned} \tag{11}$$

L.2. Empirical Evidence

Table h. Guaranteed Depth Scale Equivalence. Metric is REL.

Depth Scale	0.001×	0.1×	1×	10×	1000×
CFormer [41]	810.8	5.404	0.236	0.684	0.997
OGNI-DC [42]	7.079	0.704	0.158	0.387	0.622
G2-MD [34]	0.386	0.187	0.108	2.693	145.1
Ours	0.081	0.081	0.081	0.081	0.081

We test OMNI-DC and several baselines on the ETH3D-SfM-Indoor validation split. In each column, we multiply both the input sparse depth and ground-truth depth by a scale factor and compute the relative error:

$$\mathrm{REL}(\hat{\mathbf{D}}, \mathbf{D}^{\mathrm{gt}}) = \frac{1}{HW} \cdot \sum_{i,j}^{W,H} \frac{|\hat{\mathbf{D}}_{i,j} - \mathbf{D}^{\mathrm{gt}}_{i,j}|}{\mathbf{D}^{\mathrm{gt}}_{i,j}} \tag{12}$$

The REL error should be a constant across all scales if the model has the scale-equivariance property. Results are shown in Tab. h. Our method has the same REL error across all scales, proving the guaranteed scale equivariance in our implementation. All baselines fail catastrophically on the extreme cases (*e.g.*, $\times 1000$ when from m to mm).

M. Evaluation Details

M.1. Baselines

We run Depth Pro [4] to directly predict metric depth, without considering the sparse depth input. We estimate the global scale and shift in the least square manner against the sparse depth points for Marigold [11] (in linear depth space) and DepthAnythingv2 [40] (in disparity space).

For BP-Net [30], Depth Prompting [23], and OGNI-DC [42], we use their model trained on NYUv2 and KITTI for indoor and outdoor testing, respectively. We use the DFU [37] checkpoint trained on KITTI for all experiments, since its NYU code is not released. G2-MD [34] needs a separate scaling factor for indoors and outdoors, and we use 20.0 and 100.0 as suggested by the authors.

Note that while we provide the most favorable settings for all baselines, our method has only a *single model* and does *not* need separate hyperparameters for indoor and outdoor scenes, making it the simplest to use.

M.2. Evaluation Metrics

The metrics are defined as follows:

$$\mathrm{MAE}(\hat{\mathbf{D}}, \mathbf{D}^{\mathrm{gt}}) = \frac{1}{HW} \cdot \sum_{i,j}^{W,H} |\hat{\mathbf{D}}_{i,j} - \mathbf{D}^{\mathrm{gt}}_{i,j}|$$

$$\mathrm{REL}(\hat{\mathbf{D}}, \mathbf{D}^{\mathrm{gt}}) = \frac{1}{HW} \cdot \sum_{i,j}^{W,H} \frac{|\hat{\mathbf{D}}_{i,j} - \mathbf{D}^{\mathrm{gt}}_{i,j}|}{\mathbf{D}^{\mathrm{gt}}_{i,j}}$$

$$\mathrm{RMSE}(\hat{\mathbf{D}}, \mathbf{D}^{\mathrm{gt}}) = \sqrt{\frac{1}{HW} \cdot \sum_{i,j}^{W,H} (\hat{\mathbf{D}}_{i,j} - \mathbf{D}^{\mathrm{gt}}_{i,j})^2}$$

$$\delta_1(\hat{\mathbf{D}}, \mathbf{D}^{\mathrm{gt}}) = \frac{1}{HW} \sum_{i,j}^{W,H} \mathbf{1}\left(\max\left(\frac{\hat{\mathbf{D}}_{i,j}}{\mathbf{D}^{\mathrm{gt}}_{i,j}}, \frac{\mathbf{D}^{\mathrm{gt}}_{i,j}}{\hat{\mathbf{D}}_{i,j}}\right) < 1.25\right)$$

N. Accuracy Breakdown

More quantitative results are shown in Tabs. j to l. Compared to Tab.2 in the main paper, we separate the results for indoor and outdoor scenes. Our method works better than baselines under almost all settings.

O. Qualitative Comparison

Visualizations are provided in Figs. g to i. Compared to DC methods G2-MD [34] and OGNI-DC [42], our method generates much sharper results and is more robust to noise. While DA-v2 [40] produces sharp details, its global structure is always off, especially for outdoor scenes.

P. More Ablations on the Laplacian Loss

To show the necessity of using an L_1 loss along with L_{lap}, we conduct additional ablation studies as shown in Tab. i. Our solution with L_1 works the best. This is because DC is a dense prediction task, *i.e.*, the error on every pixel contributes to the final metrics. While L_{lap} helps convergence, it falls short of enforcing a reasonable depth for every pixel.

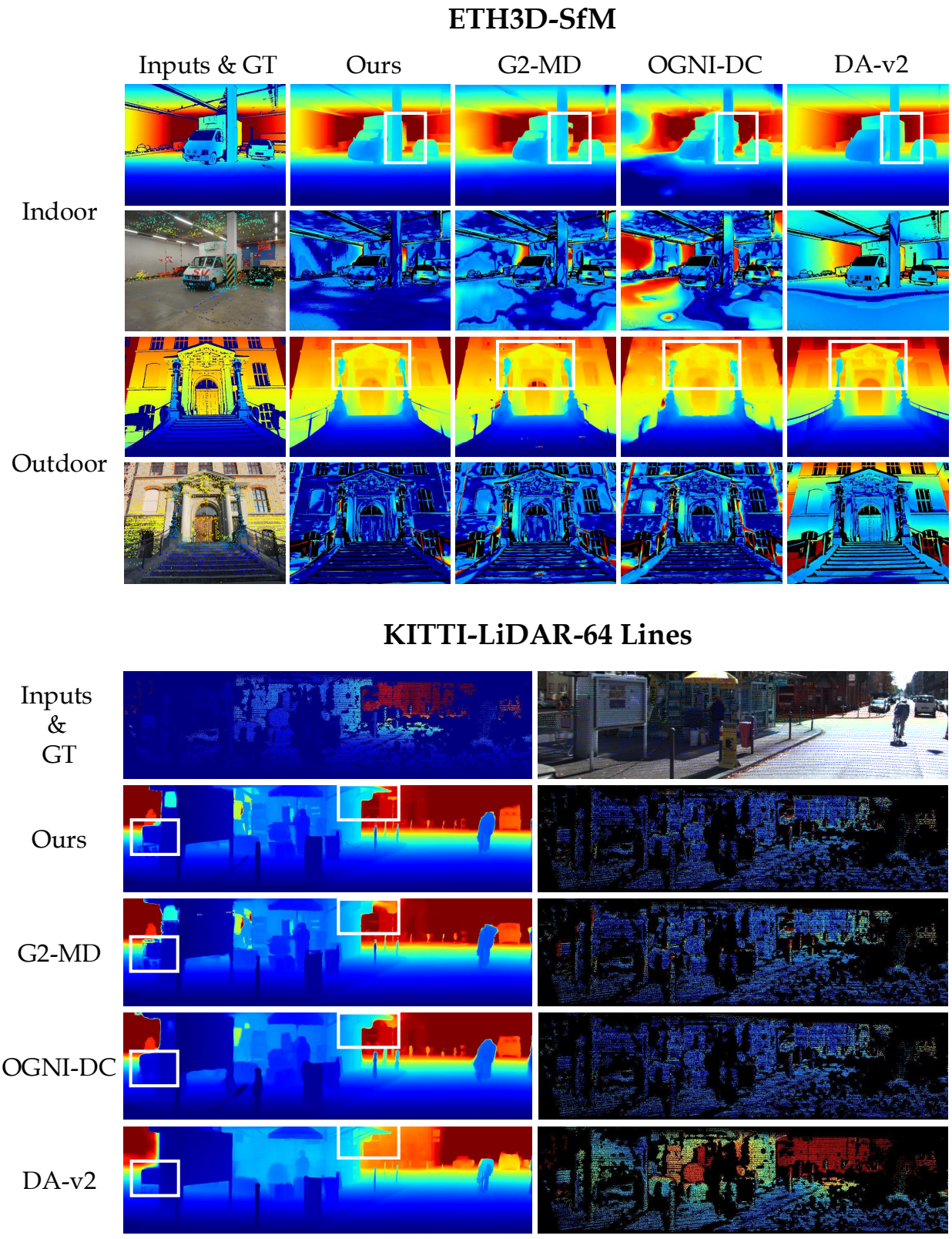

Figure g. First row/column: gt and predicted depth; second row/column: RGB, sparse depth (superimposed), and error maps (blue means small errors).

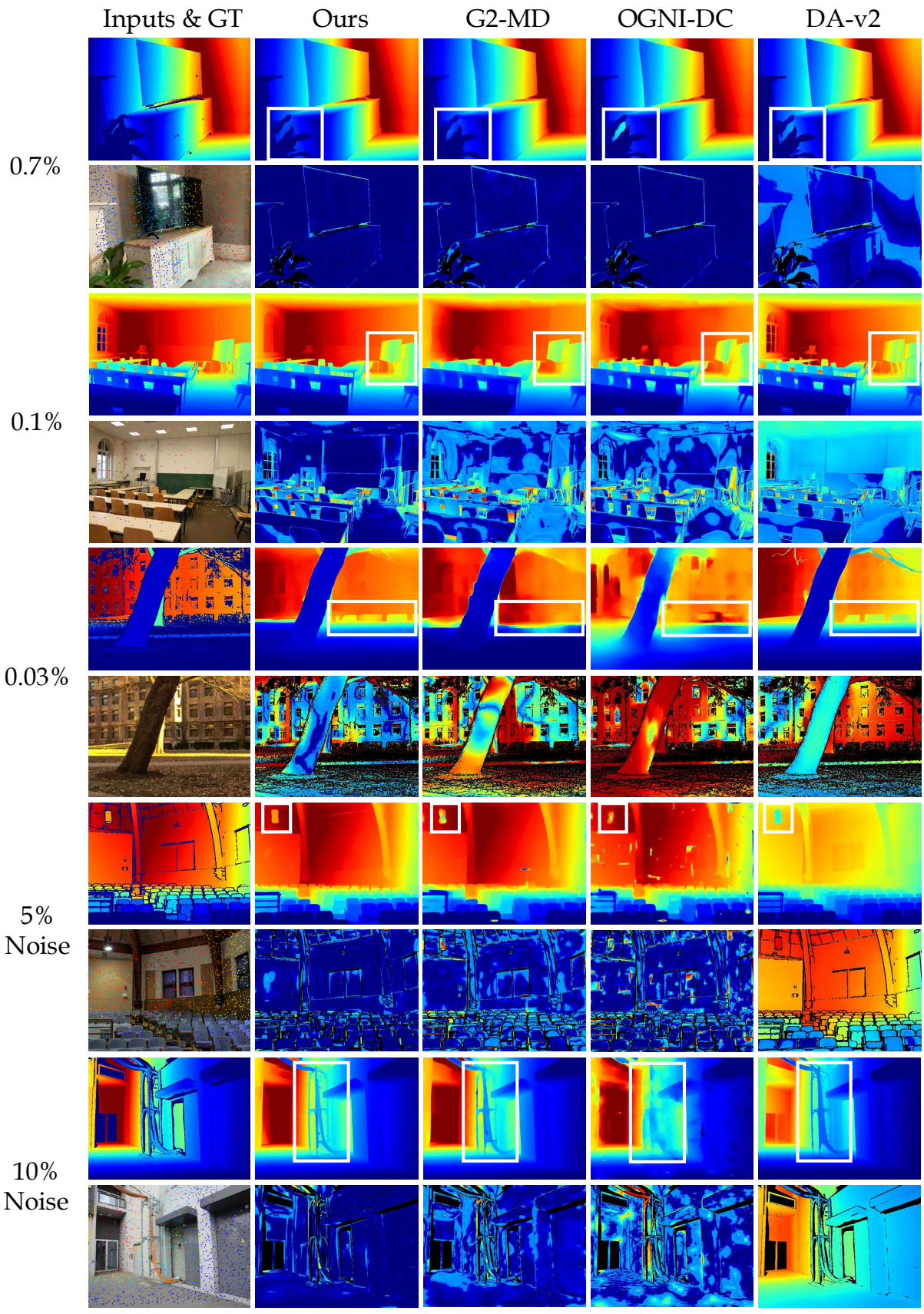

Figure h. First row: gt and predicted depth; second row: RGB, sparse depth (superimposed), and error maps (blue means small errors).

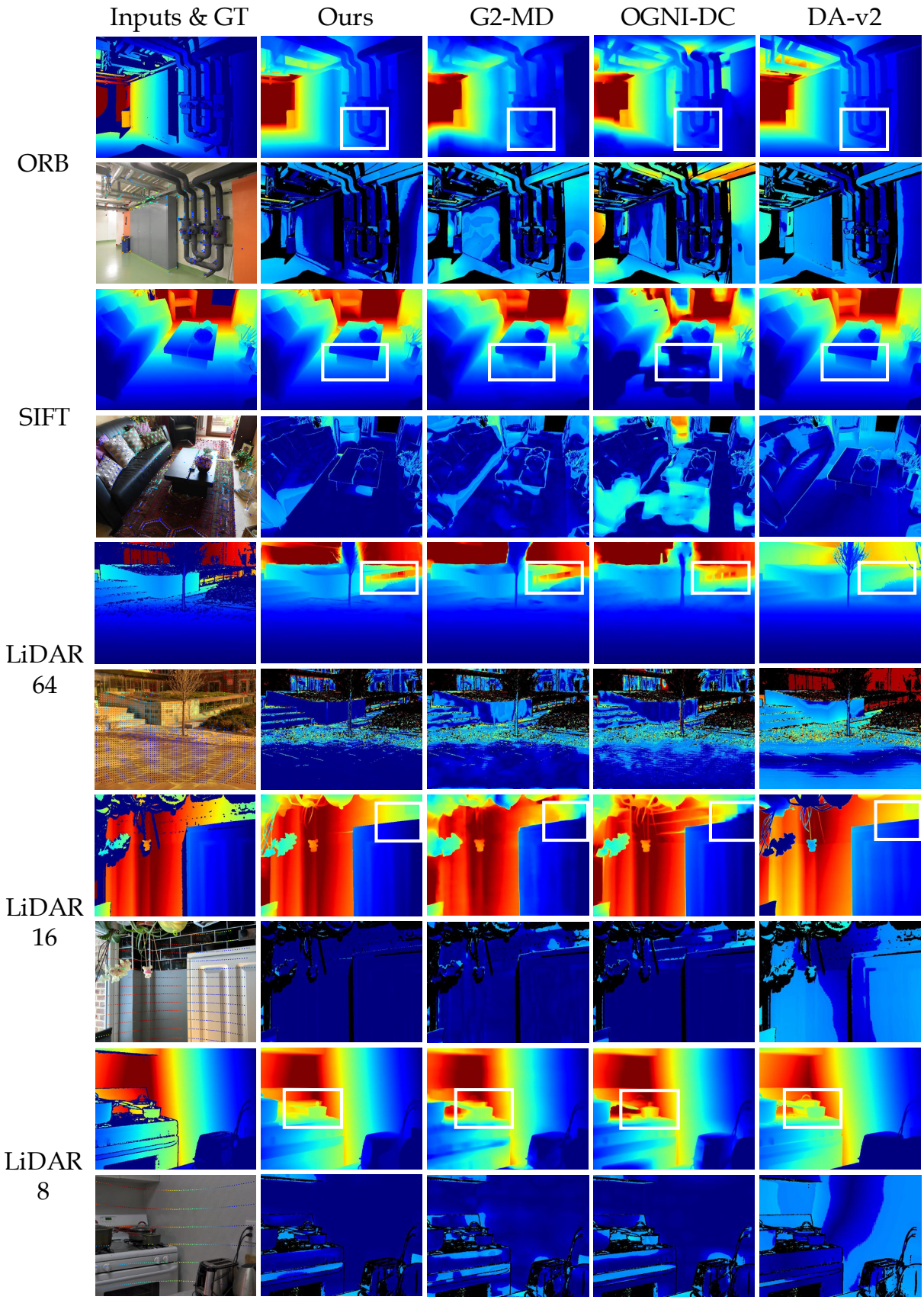

Figure i. First row: gt and predicted depth; second row: RGB, sparse depth (superimposed), and error maps (blue means small errors).

Table i. Ablations on removing the L_1 loss.

	ETH-MAE	ETH-REL	KITTI-MAE	KITTI-REL
L_{lap}	11.208	1.410	1.353	0.307
$L_{Lap} + L_{gm}$	0.525	0.081	1.179	0.282
$\boldsymbol{L_{Lap}+L_{gm}+L_1}$	**0.490**	**0.076**	**1.173**	**0.277**

Table j. Quantitative comparison with baselines on the **synthetic depth patterns** on the **indoor scenes**. Results averaged on the ARKitScenes, iBims, ETH3D-indoor, and DIODE-indoor subsets.

Methods	0.7%				0.1%				0.03%			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
Depth Pro [4]	0.636	0.524	0.176	0.746	0.636	0.524	0.176	0.746	0.636	0.524	0.176	0.746
DA-v2 [40]	0.626	0.193	0.042	0.982	0.632	0.194	0.042	0.982	0.636	0.195	0.042	0.981
Marigold [11]	0.306	0.182	0.060	0.954	0.309	0.184	0.060	0.952	0.314	0.186	0.061	0.952
CFormer [41]	0.151	0.025	0.006	0.996	0.883	0.557	0.161	0.679	1.417	1.042	0.301	0.432
DFU [37]	2.166	1.425	1.118	0.508	3.930	2.941	2.002	0.267	5.920	4.659	3.073	0.140
BP-Net [30]	0.236	0.044	0.014	0.983	0.709	0.454	0.139	0.748	1.009	0.744	0.216	0.511
OGNI-DC [42]	0.105	0.020	0.005	**0.997**	0.236	0.078	0.017	0.990	0.421	0.199	0.049	0.958
G2-MD [34]	0.107	0.024	0.007	**0.997**	0.195	0.065	0.019	0.989	0.327	0.163	0.056	0.955
Ours	**0.084**	**0.015**	**0.004**	**0.997**	**0.151**	**0.038**	**0.010**	**0.994**	**0.233**	**0.076**	**0.020**	**0.987**

Methods	5% Noise				10 % Noise				ORB [26]			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
Depth Pro [4]	0.636	0.524	0.176	0.746	0.636	0.524	0.176	0.746	0.636	0.524	0.176	0.746
DA-v2 [40]	1.079	0.527	0.217	0.857	1.793	0.851	0.339	0.701	1.507	1.123	0.797	0.963
Marigold [11]	0.318	0.190	0.063	0.954	0.347	0.217	0.072	0.949	0.426	0.311	0.131	0.893
CFormer [41]	0.253	0.056	0.017	0.983	0.335	0.096	0.031	0.965	1.420	1.059	0.339	0.415
DFU [37]	2.220	1.463	1.114	0.496	2.267	1.507	1.114	0.481	5.611	4.190	2.949	0.260
BP-Net [30]	0.315	0.089	0.030	0.964	0.393	0.142	0.050	0.939	1.228	0.906	0.354	0.422
OGNI-DC [42]	0.202	0.047	0.014	0.986	0.283	0.084	0.027	0.970	0.656	0.438	0.171	0.713
G2-MD [34]	0.134	0.029	0.008	0.996	0.155	0.034	0.009	0.995	0.438	0.280	0.124	0.824
Ours	**0.090**	**0.016**	**0.004**	**0.997**	**0.097**	**0.019**	**0.005**	**0.997**	**0.240**	**0.127**	**0.057**	**0.944**

Methods	SIFT [18]				LiDAR-64-Lines				LiDAR-16-Lines			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
Depth Pro [4]	0.636	0.524	0.176	0.746	0.636	0.524	0.176	0.746	0.636	0.524	0.176	0.746
DA-v2 [40]	0.749	0.549	0.390	**0.973**	2.359	0.300	0.108	0.980	0.597	0.189	0.041	0.982
Marigold [11]	0.413	0.301	0.127	0.905	1.166	0.182	0.060	0.954	0.306	0.182	0.060	0.954
CFormer [41]	1.315	0.978	0.317	0.442	3.473	0.017	**0.004**	**0.997**	0.255	0.075	0.020	0.981
DFU [37]	5.721	4.305	2.992	0.239	5.277	1.472	1.319	0.629	2.455	1.726	1.361	0.449
BP-Net [30]	1.150	0.836	0.328	0.469	2.217	0.037	0.012	0.985	0.346	0.110	0.036	0.954
OGNI-DC [42]	0.517	0.332	0.134	0.807	1.242	**0.016**	**0.004**	**0.997**	0.154	0.040	0.009	0.995
G2-MD [34]	0.402	0.257	0.117	0.834	0.882	0.022	0.006	**0.997**	0.150	0.045	0.012	0.994
Ours	**0.203**	**0.101**	**0.046**	0.960	**0.611**	**0.016**	**0.004**	**0.997**	**0.107**	**0.024**	**0.006**	**0.996**

Methods	LiDAR-8-Lines			
	RMSE	MAE	REL	δ_1
Depth Pro [4]	0.636	0.524	0.176	0.746
DA-v2 [40]	0.602	0.194	0.042	0.982
Marigold [11]	0.309	0.187	0.062	0.951
CFormer [41]	0.934	0.609	0.168	0.662
DFU [37]	4.022	3.029	2.141	0.257
BP-Net [30]	0.816	0.587	0.179	0.652
OGNI-DC [42]	0.287	0.114	0.028	0.979
G2-MD [34]	0.219	0.083	0.023	0.988
Ours	**0.163**	**0.050**	**0.014**	**0.993**

Table k. Quantitative comparison with baselines on the **synthetic depth patterns** on the **outdoor scenes**. Results averaged on the ETH3D-outdoor and DIODE-outdoor subsets.

Methods	0.7%				0.1%				0.03%			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
Depth Pro [4]	7.712	6.368	0.426	0.183	7.712	6.368	0.426	0.183	7.712	6.368	0.426	0.183
DA-v2 [40]	6.003	1.993	0.114	0.924	6.195	2.103	0.116	0.919	6.314	2.118	0.121	0.922
Marigold [11]	2.454	1.351	0.123	0.884	2.514	1.382	0.124	0.882	2.619	1.425	0.130	0.881
CFormer [41]	4.999	3.239	0.663	0.625	9.578	7.504	1.437	0.360	12.149	10.198	1.875	0.240
DFU [37]	2.771	1.255	0.158	0.850	5.486	3.198	0.440	0.609	7.504	4.779	0.685	0.466
BP-Net [30]	3.046	1.281	0.102	0.917	6.368	3.766	0.276	0.758	7.112	4.379	0.340	0.672
OGNI-DC [42]	1.747	0.554	0.046	0.967	2.974	1.449	0.169	0.855	4.140	2.484	0.330	0.710
G2-MD [34]	1.453	0.368	0.032	0.980	2.261	0.868	0.086	0.933	3.235	1.772	0.171	0.803
Ours	**1.275**	**0.292**	**0.022**	**0.985**	**1.889**	**0.599**	**0.044**	**0.967**	**2.477**	**0.970**	**0.070**	**0.942**

Methods	5% Noise				10 % Noise				ORB [26]			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
Depth Pro [4]	7.712	6.368	0.426	0.183	7.712	6.368	0.426	0.183	7.712	6.368	0.426	0.183
DA-v2 [40]	8.689	4.452	0.281	0.646	10.893	6.302	0.463	0.350	5.066	2.026	0.112	0.895
Marigold [11]	2.505	1.390	0.123	0.887	2.630	1.512	0.129	0.882	2.738	1.637	0.156	0.825
CFormer [41]	5.064	3.316	0.674	0.617	5.133	3.401	0.686	0.608	7.577	4.988	0.979	0.544
DFU [37]	3.262	1.620	0.185	0.800	3.713	1.995	0.213	0.747	4.376	2.469	0.370	0.655
BP-Net [30]	3.120	1.340	0.113	0.901	3.242	1.441	0.129	0.879	4.302	2.112	0.205	0.805
OGNI-DC [42]	1.962	0.690	0.057	0.954	2.160	0.822	0.069	0.940	3.019	1.480	0.194	0.826
G2-MD [34]	1.553	0.402	0.034	0.978	1.663	0.442	0.035	0.975	2.019	0.794	0.081	0.920
Ours	**1.323**	**0.313**	**0.023**	**0.983**	**1.390**	**0.341**	**0.024**	**0.982**	**1.646**	**0.514**	**0.039**	**0.967**

Methods	SIFT [18]				LiDAR-64-Lines				LiDAR-16-Lines			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
Depth Pro [4]	7.712	6.368	0.426	0.183	7.712	6.368	0.426	0.183	7.712	6.368	0.426	0.183
DA-v2 [40]	5.580	2.082	0.116	0.905	5.918	1.960	0.113	0.924	6.033	2.030	0.114	0.923
Marigold [11]	2.671	1.583	0.155	0.847	2.451	1.340	0.123	0.884	2.468	1.349	0.124	0.883
CFormer [41]	7.788	5.450	1.125	0.507	3.351	1.758	0.339	0.771	4.424	2.628	0.513	0.696
DFU [37]	4.388	2.475	0.408	0.662	2.975	1.191	0.181	0.844	3.380	1.656	0.192	0.815
BP-Net [30]	4.352	2.174	0.239	0.807	2.234	0.787	0.075	0.937	4.505	2.243	0.160	0.873
OGNI-DC [42]	2.690	1.299	0.185	0.837	1.550	0.435	0.035	0.974	2.157	0.831	0.081	0.937
G2-MD [34]	1.844	0.677	0.077	0.925	**1.200**	**0.292**	0.025	**0.985**	1.756	0.524	0.047	0.970
Ours	**1.429**	**0.403**	**0.034**	**0.974**	1.271	0.303	**0.023**	0.983	**1.513**	**0.412**	**0.031**	**0.978**

Methods	LiDAR-8-Lines			
	RMSE	MAE	REL	δ_1
Depth Pro [4]	7.712	6.368	0.426	0.183
DA-v2 [40]	6.304	2.056	0.119	0.922
Marigold [11]	2.578	1.382	0.124	0.883
CFormer [41]	7.759	5.549	1.071	0.472
DFU [37]	5.242	3.027	0.401	0.623
BP-Net [30]	5.859	3.282	0.226	0.776
OGNI-DC [42]	3.354	1.671	0.197	0.824
G2-MD [34]	2.404	0.918	0.078	0.936
Ours	**2.096**	**0.715**	**0.048**	**0.961**

Table l. Quantitative comparison with baselines on the ETH3D-SfM and KITTIDC. The numbers in gray are trained on KITTI and excluded from the ranking.

Methods	ETH3D-SfM-Indoor				ETH3D-SfM-Outdoor				KITTI-64-Lines			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
CFormer [41]	2.088	0.811	0.229	0.616	9.108	4.782	1.215	0.520	0.741	0.195	0.011	0.998
DFU [37]	3.572	2.417	1.105	0.446	4.296	2.494	0.588	0.624	0.713	0.186	0.010	0.998
BP-Net [30]	1.664	0.864	0.301	0.600	4.342	1.859	0.339	0.770	0.784	0.204	0.011	0.998
DPromting [23]	1.306	1.004	0.269	0.605	5.596	4.664	0.846	0.349	1.078	0.324	0.019	0.993
OGNI-DC [42]	1.108	0.520	0.181	0.758	2.671	1.270	0.268	0.787	0.750	0.193	0.010	0.998
Depth Pro [4]	0.928	0.749	0.208	0.659	5.433	4.824	0.441	0.196	4.893	3.233	0.211	0.651
DA-v2 [40]	**0.592**	0.280	**0.065**	**0.950**	2.663	0.805	0.082	0.935	4.561	1.925	0.090	0.924
Marigold [11]	0.627	0.472	0.152	0.842	1.883	1.270	0.252	0.715	3.462	1.911	0.118	0.889
G2-MD [34]	1.068	0.416	0.164	0.896	2.453	0.770	0.153	0.889	1.612	0.376	0.024	0.986
Ours	0.605	**0.239**	0.090	0.932	**1.069**	**0.312**	**0.053**	**0.953**	**1.191**	**0.270**	**0.015**	**0.993**

Methods	KITTI-32-Lines				KITTI-16-Lines				KITTI-8-Lines			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
CFormer [41]	1.245	0.387	0.022	0.991	2.239	0.882	0.050	0.969	3.650	1.701	0.102	0.877
DFU [37]	1.099	0.315	0.018	0.995	2.070	0.738	0.040	0.976	3.269	1.468	0.08	0.915
BP-Net [30]	1.032	0.296	0.016	0.996	1.524	0.490	0.026	0.991	2.391	0.953	0.052	0.971
DPromting [23]	1.234	0.382	0.021	0.992	1.475	0.477	0.025	0.990	1.7907	0.6344	0.0322	0.986
OGNI-DC [42]	1.018	0.268	0.014	0.996	1.664	0.453	0.022	0.990	2.363	0.777	0.039	0.977
Depth Pro [4]	4.893	3.233	0.211	0.651	4.893	3.233	0.211	0.651	4.893	3.233	0.211	0.651
DA-v2 [40]	4.583	1.928	0.090	0.923	4.615	1.934	0.090	0.923	4.689	1.951	0.091	0.922
Marigold [11]	3.463	1.902	0.117	0.892	3.468	1.904	0.117	0.891	3.498	1.939	0.120	0.885
G2-MD [34]	1.802	0.447	0.027	0.985	2.222	0.645	0.035	0.981	2.769	0.901	0.046	0.970
Ours	**1.398**	**0.339**	**0.019**	**0.990**	**1.682**	**0.441**	**0.023**	**0.987**	**2.058**	**0.597**	**0.030**	**0.982**

References

[1] Luca Bartolomei, Matteo Poggi, Andrea Conti, Fabio Tosi, and Stefano Mattoccia. Revisiting depth completion from a stereo matching perspective for cross-domain generalization. In *International Conference on 3D Vision (3DV)*, pages 1360–1370, 2024. 1

[2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *NeurIPS Datasets & Benchmarks*, 2021. 5

[3] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 2

[4] 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 preprint arXiv:2410.02073*, 2024. 4, 5, 8, 13, 14, 15

[5] Chenjie Cao, Xinlin Ren, and Yanwei Fu. Mvsformer++: Revealing the devil in transformer's details for multi-view stereo. *arXiv preprint arXiv:2401.11673*, 2024. 2

[6] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 811–820, 2024. 2

[7] Andrea Conti, Matteo Poggi, and Stefano Mattoccia. Sparsity agnostic depth completion. In *WACV*, pages 5871–5880, 2023. 1

[8] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. arxiv e-prints. *arXiv preprint arXiv:1512.03385*, 10, 2015. 3

[10] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *CVPR*, pages 2583–2592, 2021. 5

[11] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, pages 9492–9502, 2024. 5, 8, 13, 14, 15

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2

[13] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[14] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Korner. Evaluation of cnn-based single-image depth estimation methods. In *ECCV Workshops*, pages 0–0, 2018. 1, 5

[15] Han Li, Yukai Ma, Yaqing Gu, Kewei Hu, Yong Liu, and Xingxing Zuo. Radarcam-depth: Radar-camera fusion for depth estimation with learned metric scale. In *International Conference on Robotics and Automation (ICRA)*, pages 10665–10672. IEEE, 2024. 1, 2

[16] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *AAAI*, pages 1638–1646, 2022. 3

[17] Chen-Chou Lo and Patrick Vandewalle. Depth estimation from monocular images and sparse radar using deep ordinal regression network. In *ICIP*, pages 3343–3347. IEEE, 2021. 1

[18] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision (IJCV)*, 60:91–110, 2004. 3, 13, 14

[19] 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. 2

[20] 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. 3

[21] Hyoungseob Park, Anjali Gupta, and Alex Wong. Test-time adaptation for depth completion. In *CVPR*, pages 20519–20529, 2024. 1

[22] Jin-Hwi Park and Hae-Gon Jeon. A simple yet universal framework for depth completion. *Advances in Neural Information Processing Systems*, 37:23577–23602, 2025. 1

[23] Jin-Hwi Park, Chanhwi Jeong, Junoh Lee, and Hae-Gon Jeon. Depth prompting for sensor-agnostic depth estimation. In *CVPR*, pages 9859–9869, 2024. 1, 5, 8, 15

[24] 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 TPAMI*, 44(3), 2022. 3

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 3

[26] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, pages 2564–2571, 2011. 13, 14

[27] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 5

[28] Thomas Schöps, Johannes L. Schönberger, 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 *CVPR*, 2017. 2, 5

[29] Akash Deep Singh, Yunhao Ba, Ankur Sarker, Howard Zhang, Achuta Kadambi, Stefano Soatto, Mani Srivastava, and Alex Wong. Depth estimation from camera image and mmwave radar point cloud. In *CVPR*, pages 9275–9285, 2023. 1

[30] Jie Tang, Fei-Peng Tian, Boshi An, Jian Li, and Ping Tan. Bilateral propagation network for depth completion. In *CVPR*, pages 9763–9772, 2024. 5, 8, 13, 14, 15

[31] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. *WACV*, 2025. 2

[32] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017. 5

[33] 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

[34] Haotian Wang, Meng Yang, and Nanning Zheng. G2-monodepth: A general framework of generalized depth inference from monocular rgb+ x data. *IEEE TPAMI*, 2023. 1, 2, 4, 5, 8, 13, 14, 15

[35] Yufei Wang, Bo Li, Ge Zhang, Qi Liu, Tao Gao, and Yuchao Dai. Lrru: Long-short range recurrent updating networks for depth completion. In *ICCV*, pages 9422–9432, 2023. 3

[36] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *ECCV*, pages 36–54, 2024. 3

[37] Yufei Wang, Ge Zhang, Shaoqian Wang, Bo Li, Qi Liu, Le Hui, and Yuchao Dai. Improving depth completion via depth feature upsampling. In *CVPR*, pages 21104–21113, 2024. 4, 5, 8, 13, 14, 15

[38] Zhiqiang Yan, Yupeng Zheng, Deng-Ping Fan, Xiang Li, Jun Li, and Jian Yang. Learnable differencing center for nighttime depth perception. *Visual Intelligence*, 2(1):15, 2024. 2

[39] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *CVPR*, 2019. 2

[40] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *NeurIPS*, 37:21875–21911, 2024. 1, 2, 5, 6, 8, 13, 14, 15

[41] Youmin Zhang, Xianda Guo, Matteo Poggi, Zheng Zhu, Guan Huang, and Stefano Mattoccia. Completionformer: Depth completion with convolutions and vision transformers. In *CVPR*, pages 18527–18536, 2023. 1, 2, 3, 5, 8, 13, 14, 15

[42] Yiming Zuo and Jia Deng. Ogni-dc: Robust depth completion with optimization-guided neural iterations. In *ECCV*, pages 78–95, 2024. 1, 2, 3, 4, 5, 8, 13, 14, 15