

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

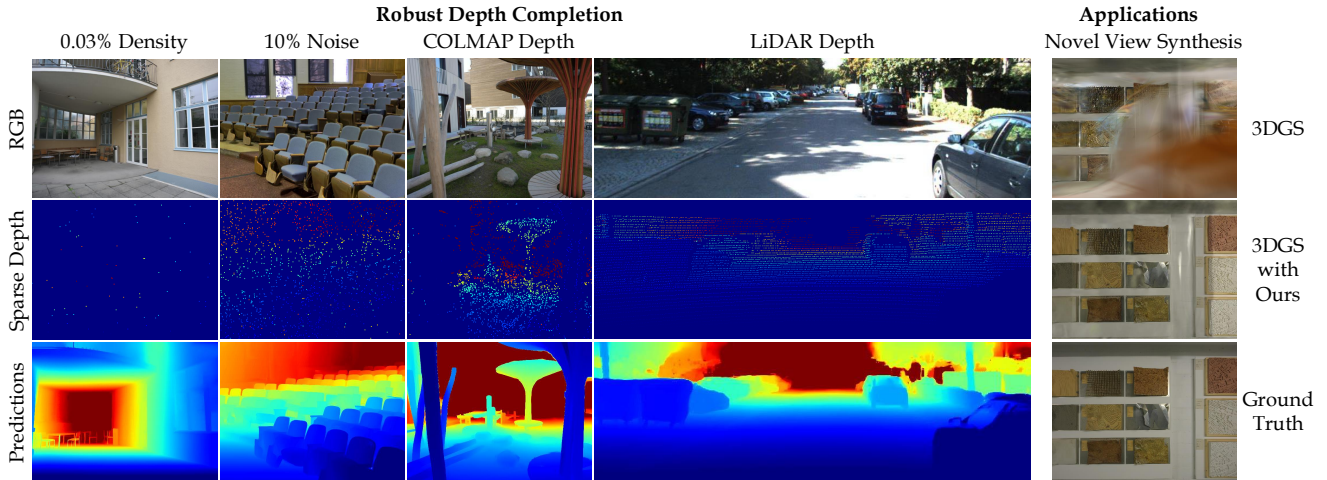


Figure 1. Given an RGB image and a sparse depth map, OMNI-DC generates high-quality dense depth maps on different types of scenes (indoors/outdoors/urban) with a variety of sparse depth patterns, in a *zero-shot* manner: none of these datasets is seen during training. The dense depth maps can be used to train 3DGS [22], greatly improving the rendering quality for novel view synthesis.

Abstract

Depth completion (DC) aims to predict a dense depth map from an RGB image and a sparse depth map. Existing DC methods generalize poorly to new datasets or unseen sparse depth patterns, limiting their real-world applications. We propose OMNI-DC, a highly robust DC model that generalizes well zero-shot to various datasets. The key design is a novel Multi-resolution Depth Integrator, allowing our model to deal with very sparse depth inputs. We also introduce a novel Laplacian loss to model the ambiguity in the training process. Moreover, we train OMNI-DC on a mixture of high-quality datasets with a scale normalization technique and synthetic depth patterns. Extensive experiments on 7 datasets show consistent improvements over baselines, reducing errors by as much as 43%. Codes and checkpoints are available at <https://github.com/princeton-vl/OMNI-DC>.

1. Introduction

Depth completion (DC) is the task of predicting a dense depth map from an RGB image and a sparse depth map. Dense and accurate depth maps are crucial for many down-

stream applications, such as robotics and 3D reconstruction, where sparse depth hints are often available. DC has been successfully applied to various tasks, including autonomous driving [8, 17], 3D reconstruction [51], and novel view synthesis [43, 54], where sparse depth data come from either active sensors such as LiDAR [8, 17, 51], or multiview matching (e.g., Structure from Motion, SfM) [43, 54].

In recent years, various methods for DC have been proposed [28, 37, 50, 60, 62, 73, 74]. While these methods achieve impressive accuracy on a single domain such as NYUv2 [34] and KITTI [55], they often fail catastrophically on unseen sparse depth patterns or new datasets [2, 13, 74]. As a result, users of downstream tasks such as view synthesis [43] or 3D reconstruction [51] have to train their own DC models on custom datasets, which is not only laborious but also could be infeasible if not enough RGB-D data are available for the test domain. This greatly limits the real-world applications of existing DC models.

This paper focuses on the most challenging and practical setting, *i.e.*, *zero-shot* generalization across different *sparsity and sensors* with a *single* model. This goal poses various challenges in the model design, due to the wide distribution of scene types and depth patterns that need to be

covered. Therefore, we propose several novel designs in our OMNI-DC framework to maximize its generalizability. The main contributions of this paper are as follows:

- We propose a novel layer named Multi-resolution Depth Integrator, and a novel Laplacian loss. The combination of them allows our method to deal with sparse depth maps of varying densities, even the extremely sparse ones.
- We propose a scale normalization method together with a pipeline for generating synthetic sparse depth patterns, which allows us to do large-scale stable training on a mixture of datasets to enhance generalization.
- Our method achieves state-of-the-art accuracy on seven datasets, reducing the error by as much as 43% compared to baselines, and can be directly applied to view synthesis.

We now provide more details on the main components.

Multi-Resolution Depth Integrator. Recent work OGNI-DC [74] shows impressive generalization capability with its Differentiable Depth Integrator design. However, a key limitation of OGNI-DC is its poor performance on extremely sparse input depth. We first study the cause of this limitation through theoretical analysis, and find that this is due to the error accumulation in the depth integration process. Based on that, we propose a Multi-resolution Depth Integrator, which allows explicit modeling of long-range depth relationships and significantly improves the performance.

Laplacian Loss. Existing DC models are trained with an L_1 or L_2 loss [73, 74]. We find that these losses are dominated by the ambiguous regions (*e.g.*, areas lacking sparse depth points), resulting in poor convergence. To resolve this issue, we propose a probability-based Laplacian loss, which allows our method to self-adaptively model the uncertainty during the training process, leading to better results.

Scale Normalization. DC models are usually trained on a single dataset (*e.g.*, NYU or KITTI), making them overfitted to a specific domain and resulting in poor generalization. While training monocular depth models on a mixture of datasets is a promising solution [41, 70], naive mixing leads to poor performance, due to the vastly different depth ranges. To this end, we introduce a novel scale normalization technique tailored to the DC task, which predicts in the log-depth space and matches the median values across samples, resulting in good convergence on all types of scenes.

Large-Scale Training with Synthetic Depth Patterns. We train OMNI-DC on 5 large-scale, high-quality datasets, covering indoor, outdoor, and urban scenes. In order to enhance the diversity of training data and to align the model with the real-world sensor distributions, we design a diverse set of synthetic depth patterns to generate the sparse depth maps, including LiDAR, SfM, and two types of noises.

State-of-the-Art Zero-Shot DC Accuracy. We conduct extensive experiments across seven real-world datasets, including the conventional DC benchmarks KITTI [55] and NYUv2 [34]. We also test on two datasets with real SfM

or VIO depth patterns (*i.e.*, ETH3D [47] and VOID [64]). Finally, since datasets with real sparse depth measurements are limited, we test on 4 standard monocular depth benchmarks with synthetically generated sparse depth patterns.

OMNI-DC outperforms all baselines by a large margin. On the outdoor split of ETH3D [47] with real SfM points, our model achieves RMSE=1.069, a 43% reduction from the second best method Marigold [21]. On KITTI with 8-lines LiDAR, our model achieves a zero-shot MAE=0.597, even better than all methods trained on KITTI.

Finally, we show a practical application of OMNI-DC on view synthesis with sparse input views. We train 3DGS [22] with an auxiliary depth loss [54]. The rendering quality improves greatly compared to the vanilla 3DGS (PSNR=20.38 vs 15.64) or using other depth supervisions [4, 57].

2. Related Work

Depth Completion Models. In recent years, various deep-learning-based models have been proposed to tackle the DC task. Liu *et al.* [29] introduces the spatial propagation network (SPN), which iteratively propagates the initial predictions to its neighboring pixels through a set of learned combination weights. Different SPN variants have been proposed [9, 10, 28, 30, 37, 66]. LRRU [60] and DFU [62] first use heuristics-based algorithms to pre-fill the depth maps and then learn to refine the initial predictions. BP-Net [50] employs a learned pre-processing stage to directly propagate the sparse depth points. Other works focus on improving the neural network architecture for DC models. CompletionFormer [73] proposes a hybrid Transformer-CNN backbone suitable for the DC task. OGNI-DC [74] first predicts depth gradients with a ConvGRU and uses a depth integration layer (DDI) to convert it into depth, and then performs iterative updates. Our model is based on OGNI-DC, but we propose novel designs including Multi-res DDI and the Laplacian loss, which significantly improve the performance of OGNI-DC on extremely sparse depth inputs. Moreover, all these methods focus on overfitting to a single type of sensor on a single dataset, in contrast to the dataset and sensor agnostic generalizable setting of our method.

Generalization Across Depth Patterns. Several works focus on sparsity-agnostic generalization. SpAgNet [13] proposes a depth scaling and substitution module and can deal with very sparse depth inputs. Xu *et al.* [65] utilizes a guidance map from a monocular depth network. VPP4DC [2] finetunes a stereo matching network on a synthetic dataset by projecting the sparse depth points as virtual mosaic patterns onto the images. Depth Prompting [39] starts from a monocular depth network and learns affinities from the sparse depth map for value propagation. While these methods can generalize across different sparsity, they are trained on a single dataset and cannot generalize beyond the training domain. In contrast, our OMNI-DC is trained on a mix-

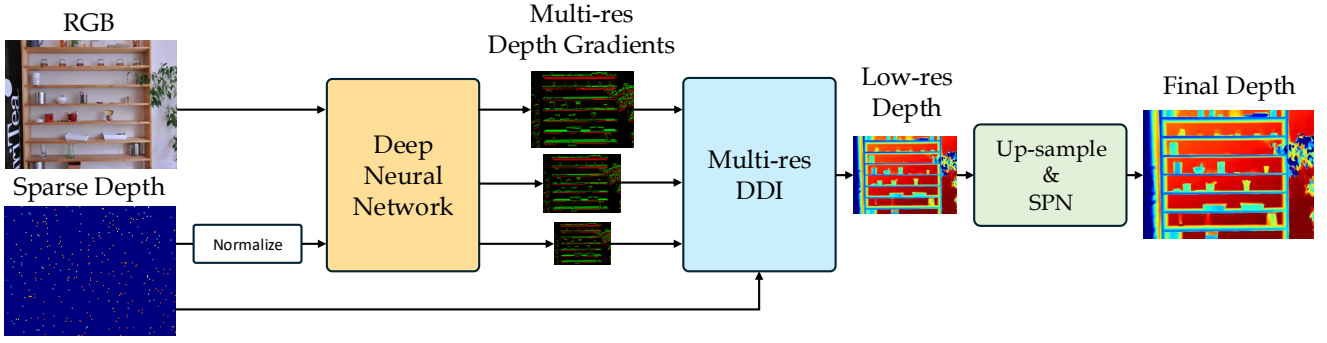


Figure 2. The overall pipeline of OMNI-DC. The RGB image and the normalized sparse depth (Sec. 3.5) are fed into a neural network to produce a set of multi-resolution depth gradient maps. These depth gradient maps are integrated into a dense depth map with the Multi-resolution DDI (Sec. 3.3). Finally, the dense map is up-sampled and processed by an SPN [28] to produce the final prediction (Sec. 4.2).

ture of data and can generalize zero-shot to new datasets.

Generalization Across Datasets. Previous works use different settings for cross-dataset generalization. UniDC [38] focuses on the few-shot setting (*i.e.*, 1-100 labeled images), and utilizes depth foundation models with hyperbolic space for fast adaption. TTADC [36] utilizes test-time adaptation (which requires a lot of unlabeled images) to close the domain gaps. Our *zero-shot* setting is strictly more challenging, and also more practical because no data need to be collected from the target domain for fine-tuning.

More recently, G2-MonoDepth (G2-MD) [57] also focuses on the zero-shot setting and proposes to jointly solve monocular depth estimation and depth completion with one model by using a unified loss. However, G2-MD only tests with random sparse depth samples, which doesn't match the distribution of sensors in the real world. We do a large-scale comparison with G2-MD and show that our method consistently performs better than G2-MD on all datasets.

Generalizable Depth Estimation. Various models [4, 6, 15, 21, 69, 70] have been proposed for generalizable depth estimation since MiDaS [41]. Depth Anything v2 [69, 70] trains on pseudo-labels generated on unlabeled real images. Depth Pro [6] proposes a two-stage training strategy to first train on real images and then only on synthetic images for finer details. Marigold [21] and GeoWizard [15] use pre-trained diffusion models as a strong prior. Compared to them, our OMNI-DC simplifies the training pipeline by training purely on synthetic images from scratch, but generalizes surprisingly well to real-world benchmarks. Furthermore, we show that the existing monocular depth models cannot effectively utilize the sparse depth information, and work poorly on the DC benchmarks.

3. Methods

We define the task of depth completion as follows: the model takes an RGB image $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$, a sparse depth observation map $\mathbf{O} \in \mathbb{R}_+^{H \times W}$, and a valid mask $\mathbf{M} \in \{0, 1\}^{H \times W}$ as input. \mathbf{M} specifies the pixels with valid depth values in \mathbf{O} . The model outputs a dense depth map with the same resolution as input: $\hat{\mathbf{D}} \in \mathbb{R}_+^{H \times W}$. In the rest

of this paper, we use hat ($\hat{\cdot}$) to denote the predicted values.

The overall pipeline of our method is shown in Fig. 2. In the rest of this section, we introduce OGNI-DC and its limitations (Secs. 3.1 and 3.2), the Multi-res DDI design (Sec. 3.3), the Laplacian loss (Sec. 3.4), the scale normalization (Sec. 3.5), and the training paradigm (Sec. 3.6).

3.1. Preliminaries: OGNI-DC

OGNI-DC [74] begins by using a deep neural network F to predict a depth gradient map $\hat{\mathbf{G}}$. $\hat{\mathbf{G}}$ models the depth relationships between neighboring pixels (see Fig. 3.III):

$$\hat{\mathbf{G}} = [\hat{\mathbf{G}}^x, \hat{\mathbf{G}}^y] = F(\mathbf{I}, \mathbf{O}; \theta), \quad (1)$$

where θ is the parameters of the neural network; x and y are the horizontal and the vertical direction, respectively.

The key component of OGNI-DC is a parameter-free custom layer named Differentiable Depth Integrator (DDI). DDI takes the depth gradient map and the sparse depth map as input, and outputs a dense depth map. This is achieved by solving a linear least squares problem involving constraints from both the sparse depths and the depth gradients:

$$\hat{\mathbf{D}} = \arg \min_{\mathbf{D}} \left(\alpha \cdot \mathcal{E}_O(\mathbf{D}, \mathbf{O}, \mathbf{M}) + \mathcal{E}_G(\mathbf{D}, \hat{\mathbf{G}}) \right), \quad (2)$$

where α is a hyperparameter, and

$$\begin{aligned} \mathcal{E}_O &:= \sum_{i,j}^{W,H} \mathbf{M}_{i,j} \cdot (\mathbf{D}_{i,j} - \mathbf{O}_{i,j})^2, \\ \mathcal{E}_G &:= \sum_{i,j}^{W,H} \left(\mathbf{G}_{i,j}^x - \hat{\mathbf{G}}_{i,j}^x \right)^2 + \left(\mathbf{G}_{i,j}^y - \hat{\mathbf{G}}_{i,j}^y \right)^2, \end{aligned} \quad (3)$$

with i, j being the pixel index; \mathbf{G}^x and \mathbf{G}^y being the finite differences: $\mathbf{G}_{i,j}^x := \mathbf{D}_{i,j} - \mathbf{D}_{i-1,j}$; $\mathbf{G}_{i,j}^y := \mathbf{D}_{i,j} - \mathbf{D}_{i,j-1}$.

Intuitively, \mathcal{E}_O encourages the predicted depth to be consistent with the observed depth at valid locations, and \mathcal{E}_G fills the missing areas with the learned depth gradients. DDI can be loosely understood as an integration process from known pixels to unknown ones. DDI alleviates the need for the neural network to learn an identity mapping for known pixels, thereby providing a strong inductive bias.

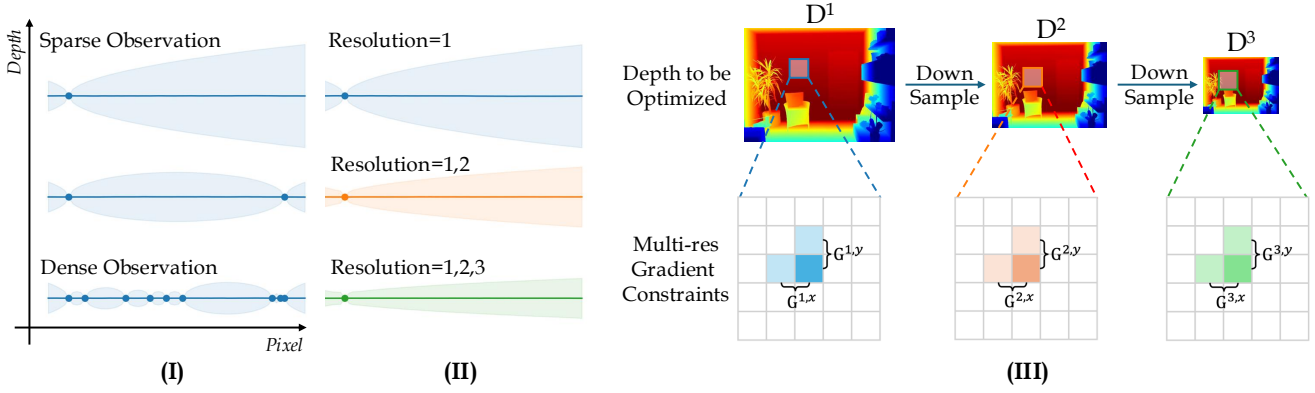


Figure 3. The Multi-resolution DDI reduces the error accumulation in the depth integration. In (I) and (II) we mark the pixels with known depth in dots, and show the mean and the 95% confidence interval of the integrated depth, assuming the ground-truth depth gradients are 0, and an i.i.d. Gaussian noise on their predictions. (I): Noise accumulates in the integration process, especially obvious when the known depths are sparse. (II): Our Multi-resolution DDI reduces the error, as it explicitly models the long-range depth dependencies. (III): Multi-res DDI is implemented by down-sampling the optimization target depth map and computing the finite difference at each resolution.

3.2. Limitation of DDI on Extremely Sparse Depth

While OGNI-DC achieves good generalization, it performs poorly when the depth observations are extremely sparse, *e.g.*, only 5 points on NYUv2 [74]. This limitation causes problems in real-world applications: when the sparse depths are obtained from SfM, the texture-less surfaces often have no reliable correspondence (Fig. 4). Similarly, active sensors often fail to generate depth on transparent or metallic surfaces, leaving large blank areas in the depth maps [1, 20].

We examine the cause of this limitation, and find that it is due to the error accumulation in the long-range integration. To illustrate this, we simplify the problem into 1D and assume an i.i.d. Gaussian additive noise with variance σ^2 on the network’s depth gradient prediction at pixel i :

$$\hat{\mathbf{G}}_i = \mathbf{G}_i^{\text{gt}} + \mathbf{n}_i, \mathbf{n}_i \sim \mathcal{N}(0, \sigma^2). \quad (4)$$

Assuming we only know the depth at pixel location 0, \mathbf{D}_0 , the predicted depth at location n is obtained by integrating the gradient values from 0 to n :

$$\hat{\mathbf{D}}_n = \mathbf{D}_0 + \sum_{i=1}^n \hat{\mathbf{G}}_i \sim \mathcal{N}(\mathbf{D}_0 + \sum_{i=1}^n \mathbf{G}_i^{\text{gt}}, n \cdot \sigma^2). \quad (5)$$

The variance of $\hat{\mathbf{D}}_n$, $n \cdot \sigma^2$, increases linearly w.r.t. the distance to the nearest known pixel. It implies that the neural network’s prediction error accumulates in the integration process, and the depth predictions are sensitive to the error in the depth gradient predictions when modeling long-range relationships. As illustrated in Fig. 3.I, when the observed depth map is relatively dense, the error accumulation is negligible. However, when the observed depths become sparser, the regions far from the observations become under-constrained and have a high depth prediction error. This explains why OGNI-DC performs badly on extremely sparse inputs.

3.3. Multi-Resolution Depth Integration

We propose a simple yet effective solution to enable DDI to overcome this limitation. We formulate a multi-resolution integration process, which jointly considers the depth relationships across multiple scales, reducing the integration error over long distances. Formally, we adjust the network to predict a set of depth gradient maps at different scales, where the resolution of each scale is different by a factor of 2, and R is the total number of resolutions:

$$\{\hat{\mathbf{G}}^r\}_{r=1, \dots, R} = F(\mathbf{I}, \mathbf{O}; \theta), \hat{\mathbf{G}}^r \in \mathbb{R}^{2 \times H/2^{r-1} \times W/2^{r-1}}. \quad (6)$$

We then extend the original DDI to incorporate multi-resolution depth gradients. Denote \mathbf{D} to be the depth map to be optimized in Eq. (2). We down-sample \mathbf{D} with a set of average-pooling layers:

$$\mathbf{D}^r = \text{AvgPool2D}(\mathbf{D}, 2^{r-1}), r = 1, \dots, R. \quad (7)$$

We modify \mathcal{E}_G in Eq. (3) and define the multi-resolution depth gradients energy term as

$$\mathcal{E}_G^R := \sum_{r=1}^R \sum_{i,j}^{W,H} \left(\mathbf{G}_{i,j}^{r,x} - \hat{\mathbf{G}}_{i,j}^{r,x} \right)^2 + \left(\mathbf{G}_{i,j}^{r,y} - \hat{\mathbf{G}}_{i,j}^{r,y} \right)^2, \quad (8)$$

where $\mathbf{G}_{i,j}^{r,x} := \mathbf{D}_{i,j}^r - \mathbf{D}_{i-1,j}^r$; $\mathbf{G}_{i,j}^{r,y} := \mathbf{D}_{i,j}^r - \mathbf{D}_{i,j-1}^r$. Finally, we solve the linear least squares following Eq. (2) to get the layer output $\hat{\mathbf{D}}$.

The computation of the multi-resolution constraints is illustrated in Fig. 3.III, and the benefit is demonstrated in Fig. 3.II: the error bound of the integrated depth is reduced greatly in the extremely sparse input case when using 3 resolutions, compared to the vanilla DDI with 1 resolution. Intuitively, Multi-resolution DDI achieves a better modeling of the global structure, as the steps required for integration are reduced from n to $n/2^{R-1}$ for a pixel n distance away from the nearest known pixel, and the local details are still preserved by the constraints at the finer resolutions.

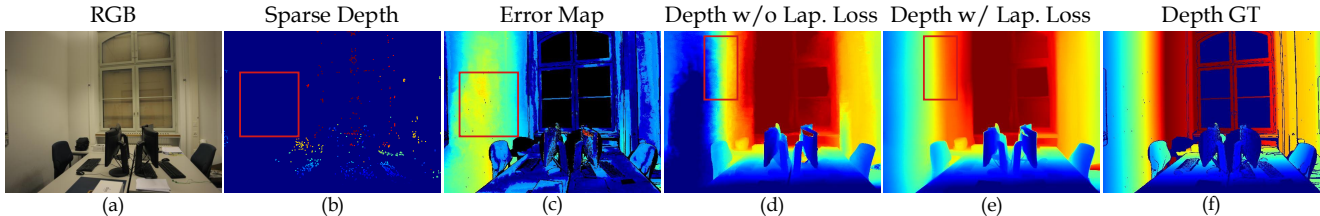


Figure 4. The sparse depth map from COLMAP [46] often has large holes on the textureless surfaces (b). These areas with high ambiguity dominate the L_1 training error (c). As a result, the model trained without the Laplacian loss does not converge well, producing artifacts in the depth map (d). In contrast, our model with the Laplacian loss generates a high-quality, smooth depth map (e).

Note that the number of constraints decreases exponentially as the resolution increases. Therefore, the additional computation overhead is marginal compared to the vanilla DDI. A comparison of the inference speed and the parameter count of OMNI-DC against baselines is shown in Fig. 6. Our method is $2\times$ faster than OGNI-DC in inference.

3.4. Laplacian Loss

As shown in Fig. 4 (b), the sparse depth maps often contain large blank areas (“holes”) with missing depth observation. Models typically make much larger errors in these areas due to the high ambiguity of depth values (Fig. 4 (c)).

Therefore, when trained with an L_1 loss, the model focuses on optimizing the high-ambiguity regions to capture the global structure accurately. The local details are thus not well optimized, leaving artifacts in the predicted depth maps, as shown in Fig. 4 (d).

We propose to use a probability-based loss to explicitly model the uncertainty of the depth prediction to achieve a smoother result. Specifically, rather than predicting a single depth value, the model predicts the mean $\hat{\mathbf{D}}$ and a per-pixel scale parameter b of the Laplacian distribution, and we use its negative log-likelihood as the Laplacian loss:

$$L_{Lap}(\mathbf{D}^{gt}, \hat{\mathbf{D}}, b) = \log(2b) + |\mathbf{D}^{gt} - \hat{\mathbf{D}}|/b. \quad (9)$$

L_{Lap} can be viewed as a probabilistic variant of L_1 . We don’t use an L_2 or Gaussian loss because we find L_2 more sensitive to outliers and make training less stable.

Although training with L_{Lap} alone produces smoother results, it reduces the model’s ability to handle noise in the sparse depth map, as it can cheat by predicting large uncertainties (See Appendix Sec. P). We find that combining L_{Lap} with the L_1 loss yields the best results. We also adopt the gradient-matching loss L_{gm} proposed in the monocular depth estimation [27, 41, 69] on the predicted depth map.

The final loss can be written as:

$$L = L_1 + 0.5 \cdot L_{Lap} + 2.0 \cdot L_{gm}. \quad (10)$$

While probability-based losses have been used in previous works on various tasks [5, 26, 49, 53, 61], we are the first to apply it to the task of depth estimation or depth completion, and prove its usefulness through experiments.

3.5. Scale Normalization

We desire our DC model to work well across a large variety of scenes, which have a large variation in the depth scale,

e.g., $< 1m$ for indoors and $> 100m$ for urban scenes.

Several problems will occur if we naively process depth in the metric space like previous DC methods [73, 74]. 1) *Model Capacity*: since the sparse depth map is part of the neural network input, the network has to learn to process a wide value range, posing challenges to the network capacity. 2) *Unbalance Among Datasets*: the commonly used L_1 or L_2 losses incur a larger penalty on larger depth values when the relative errors are the same. Therefore, the training loss focuses more on outdoor scenes, which is undesirable. 3) *Scale Ambiguity in SfM*: SfM algorithms can only reconstruct scenes up to an arbitrary global scale [46]. Recovering metric depth is often impossible and sometimes unnecessary for applications such as view synthesis [43].

In monocular depth estimation, the scale ambiguity is often resolved by using the scale-invariant loss [41, 69, 70]. However, this won’t work in DC, because instead of allowing the model to choose an arbitrary scale to predict, we require the scale of the output depth to be *equivariant* to the scale of the input sparse depth in DC. For an arbitrary scale factor β , scale equivariance is formally defined as:

$$\hat{\mathbf{D}}(\mathbf{I}, \beta \cdot \mathbf{O}) = \beta \cdot \hat{\mathbf{D}}(\mathbf{I}, \mathbf{O}), \forall \beta \in \mathbb{R}_+. \quad (11)$$

To address the scale issue, we propose a scale normalization technique tailored to DC. First, we convert all depth into the log space, where the arbitrary multiplicative scale factor becomes additive, making it suitable for the linear formulation of DDI. Second, we normalize the input sparse depth map by its median value, so that the value ranges are matched between the indoor and outdoor scenes:

$$\hat{\mathbf{G}} = F(\mathbf{I}, \tilde{\mathbf{O}}; \theta), \tilde{\mathbf{O}} = \log(\mathbf{O}) - \log(\text{median}(\mathbf{O})). \quad (12)$$

As illustrated in Fig. 2, only the network input is normalized, but *not* the sparse depth used in DDI. Therefore, the original scale of \mathbf{O} is preserved in the final output through DDI, and our model achieves *guaranteed* scale equivariance. See Appendix Sec. L for proofs.

3.6. Large-Scale Training with Synthetic Patterns

We train our model on a collection of 5 synthetic datasets, covering indoor, outdoor, and urban scenes, with a total of 573K images. The details of the datasets used for training are shown in Tab. 1. We choose synthetic datasets because real-world datasets with high-quality depth ground-truth are very limited, and we find that mixing real-world datasets

(e.g., NYU) for training produces blurry results, as shown in Appendix Sec. B. While we don’t use the complicated two-stage strategy mixing unlabeled real-world data [6, 70], we find that synthetic data alone yields surprisingly good results on real-world benchmarks.

During training, the sparse depth maps are synthetically generated by sub-sampling the dense ground-truth. Previous works such as G2-MD [57] use random samples, which align poorly with the sparse point distributions of the real sensors. We instead design two kinds of synthetic sparse patterns: 1) *SfM*: sparse points are sampled at the SIFT [32] keypoints. 2) *LiDAR*: we simulate a random 4-128 lines LiDAR with angle and shift variations.

Additionally, we simulate two types of noise for generating the sparse depth map. 1) *Outliers*: this is common in the COLMAP output due to mismatched keypoints. We simulate the outliers by randomly sampling depth values within the scene depth bounds. 2) *Boundary Noise*: blended foreground and background depth points near object boundaries occur due to viewpoint differences between the LiDAR and RGB camera have been observed in [12, 40]. We simulate it by projecting the depth map to a virtual neighboring view, inpainting the holes, sampling sparse depth, and projecting back. See Appendix Sec. F for more details.

4. Experiments

4.1. Evaluation Datasets

We test on 7 real-world datasets to show the effectiveness of our method, as shown in Tab. 1. For NYUv2, we use the standard test set and cropping following [73, 74] and test on two different densities (500/50 points). For KITTI, we use its official validation set and follow [19] to sub-sample the original 64-lines lidar into sparser inputs. We use the original test set of VOID [64] with 3 densities (1500/500/150).

To test the model’s capability on dealing with SfM inputs, we use the ETH3D [47] dataset and project the COLMAP SfM sparse depth points into the image space to get sparse depth maps. We call this dataset ETH3D-SfM.

Finally, since the datasets with sparse depth from real sensors are limited, we utilize 4 benchmarks commonly used in monocular depth estimation (iBims, ARKitScenes, ETH3D, DIODE) and generate sparse depth patterns synthetically. We sample random points with different densities [0.7%/0.1%/0.03%]; add [5%/10%] noise on top of the 0.7% density; sample depth at the [SIFT [32]/ORB [45]] keypoints; construct synthetic LiDAR points with [64/16/8] scanning lines. This synthetic subset helps us understand how these factors alone affect the model performance.

4.2. Implementation Details

We use CompletionFormer [73] as the backbone, and 3 resolutions for the DDI. The DDI generates an intermediate

Table 1. Detailed statistics of the datasets used in this paper. [†]ARKitScenes [3] has 450K images in total. We randomly sample 800 images from its validation split for faster evaluation.

Split	Dataset Name	Size	Sparse Depth	Scene Type
Training	Hypersim [42]	66K	Synthetic (Sec. 3.6)	Indoor
	IRS [58]	60K		Indoor
	Tartanair [59]	307K		In/Out
	BlendedMVS [71]	115K		Misc
	Virtual KITTI [16]	25K		Urban
Testing	iBims [24]	100	Synthetic	Indoor
	ARKitScenes [3]	800 [†]	Synthetic	Indoor
	VOID [64]	2400	SfM	Indoor
	NYUv2 [34]	652	Synthetic	Indoor
	DIODE [56]	771	Synthetic	In/Out
	ETH3D [47]	454	SfM	In/Out
	KITTIDC [55]	1000	LiDAR	Urban

depth map at the 1/4 resolution, which is refined by a convex up-sampling layer [52] and a DySPN [28], following OGNI-DC [74]. Compared to [74], we remove the iterative updates with ConvGRU because we find it not helpful for performance when trained on large-scale datasets.

We train OMNI-DC on 10×48GB GPUs, with an effective batch size of 60. In each epoch, we randomly sample 25K images from each dataset. The model is trained for 72 epochs, which takes about 6 days in total. Additional details are provided in Appendix Sec. F.

4.3. Baselines

We compare against state-of-the-art DC baselines CompletionFormer [73] (CFormer), DFU [62], BP-Net [50], OGNI-DC [74], Depth Prompting [39] (DPrompting), and G2-MonoDepth [57] (G2-MD). We also compare against the generalizable metric depth (Depth Pro [6]) and affine-invariant depth models (DepthAnythingv2 [70] (DAv2) and Marigold [21]), for which we estimate the global scale and shift under the best alignment with the sparse depth.

4.4. Results on KITTIDC and NYUv2

Results are shown in Tab. 3. While tested zero-shot, OMNI-DC even outperforms all DC methods trained on KITTI in terms of MAE (MAE=0.597 vs 0.634 for DPrompting [74]) on the 8-lines input. On the 64-lines input, ours works much better than all other methods tested zero-shot (RMSE=1.191 vs 1.612 for G2-MD [57], a 26% reduction).

On NYUv2 (Tab. 3), OMNI-DC works better than all other zero-shot methods on different densities, achieving comparable performance with models trained on NYU (500 points, RMSE=0.111 vs 0.089 for OGNI-DC [74]).

4.5. Results on the VOID and ETH3D-SfM

VOID [64] has ground-truth depth collected with an Intel RealSense camera, and the sparse depths come from a visual-inertial odometry system with three different sparsity levels. As shown in Tab. 5, our method outperforms all

Table 2. Large-scale testing on diverse datasets with synthetic sparse depth patterns. The 1st, 2nd, 3rd place methods are marked accordingly. Results are averaged on 4 datasets (Sec. 4.1). Definitions of the sparse depth patterns can be found in Sec. 4.1.

Methods	0.7%		0.1%		0.03%		5% Noise		10% Noise	
	RMSE	REL	RMSE	REL	RMSE	REL	RMSE	REL	RMSE	REL
Depth Pro [6]	0.938	0.259	0.938	0.259	0.938	0.259	0.938	0.259	0.938	0.259
DepthAnythingv2 [70]	0.818	0.066	0.834	0.067	0.845	0.068	1.299	0.238	1.922	0.380
Marigold [21]	0.367	0.081	0.373	0.082	0.384	0.084	0.379	0.083	0.406	0.091
CompletionFormer [73]	0.434	0.225	1.227	0.586	1.755	0.826	0.506	0.236	0.565	0.249
DFU [62]	1.629	0.798	2.986	1.481	4.447	2.277	1.697	0.805	1.759	0.813
BP-Net [50]	0.361	0.044	0.898	0.185	1.147	0.257	0.418	0.058	0.478	0.076
OGNI-DC [74]	0.187	0.018	0.355	0.068	0.557	0.143	0.265	0.029	0.333	0.041
G2-MonoDepth [57]	0.168	0.015	0.280	0.041	0.434	0.094	0.193	0.016	0.214	0.018
Ours	0.135	0.010	0.211	0.020	0.289	0.034	0.141	0.010	0.147	0.011

Methods	ORB [45]		SIFT [32]		LiDAR-64-Lines		LiDAR-16-Lines		LiDAR-8-Lines	
	RMSE	REL	RMSE	REL	RMSE	REL	RMSE	REL	RMSE	REL
Depth Pro [6]	0.938	0.259	0.938	0.259	0.938	0.259	0.938	0.259	0.938	0.259
DepthAnythingv2 [70]	1.343	0.569	0.871	0.299	0.974	0.110	0.800	0.065	0.822	0.068
Marigold [21]	0.467	0.140	0.453	0.136	0.367	0.081	0.369	0.082	0.378	0.082
CompletionFormer [73]	1.452	0.553	1.396	0.586	0.303	0.116	0.465	0.184	1.140	0.469
DFU [62]	4.033	2.090	4.107	2.130	2.003	0.940	1.862	0.971	3.031	1.561
BP-Net [50]	1.105	0.304	1.057	0.299	0.296	0.033	0.531	0.078	0.935	0.195
OGNI-DC [74]	0.639	0.179	0.524	0.151	0.162	0.014	0.247	0.033	0.415	0.085
G2-MonoDepth [57]	0.427	0.110	0.391	0.104	0.143	0.012	0.217	0.024	0.306	0.042
Ours	0.247	0.045	0.211	0.037	0.121	0.008	0.164	0.014	0.231	0.023

Table 3. Results on ETH-3D, KITTI, and NYUv2. Numbers in gray are trained on KITTI/NYU, and are thus excluded from the ranking.

Methods	ETH-SfM-Out		ETH-SfM-In			KITTI-64Line		KITTI-8Line		NYU-500Pt		NYU-50Pt	
	RMSE	REL	RMSE	REL		RMSE	MAE	RMSE	MAE	RMSE	REL	RMSE	REL
CFormer [73]	9.108	1.215	2.088	0.229	Trained on KITTI/NYU	0.741	0.195	3.650	1.701	0.090	0.012	0.707	0.181
DFU [62]	4.296	0.588	3.572	1.105		0.713	0.186	3.269	1.468	0.091	0.011	-	-
BP-Net [50]	4.342	0.339	1.664	0.301		0.784	0.011	2.391	0.953	0.089	0.012	0.609	0.157
DPrompting [39]	5.596	0.846	1.306	0.269		1.078	0.324	1.791	0.634	0.105	0.015	0.213	0.043
OGNI-DC [74]	2.671	0.268	1.108	0.181		0.750	0.193	2.363	0.777	0.089	0.012	0.207	0.038
Depth Pro [6]	5.433	0.441	0.928	0.208		4.893	3.233	4.893	3.233	0.266	0.062	0.266	0.062
DA-v2 [70]	2.663	0.082	0.592	0.065	Zero-shot	4.561	1.925	4.689	1.951	0.309	0.061	0.330	0.063
Marigold [21]	1.883	0.252	0.627	0.152		3.462	1.911	3.498	1.939	0.426	0.115	0.436	0.118
G2-MD [57]	2.453	0.153	1.068	0.164		1.612	0.376	2.769	0.901	0.122	0.017	0.286	0.056
Ours	1.069	0.053	0.605	0.090		1.191	0.270	2.058	0.597	0.111	0.014	0.225	0.041

baselines by a large margin for denser inputs (1500/500), and is very close to DAv2 [70] on the 150 points subset.

For ETH3D-SfM, the sparse depth is from COLMAP. As shown in Tab. 3, OMNI-DC significantly outperforms all baselines on the ETH3D outdoor split (RMSE=1.069 vs 1.883 for Marigold, a 43% reduction). On the indoor split, our method works better than all DC baselines.

4.6. Results on Synthetic Depth Patterns

Results are shown in Tab. 2. We divide the RMSE on outdoor scenes by 5.0 to make the scale approximately match with indoors, and we report separated numbers in Appendix Sec. N. OMNI-DC outperforms all baselines by a large margin on all depth patterns: it continues to work well with extremely sparse points (0.03%, REL=0.034 vs 0.068 for DAv2 [70]), a large proportion of noise (10% noise, RMSE=0.147 vs 0.214 for G2-MD [57]), or when the sparse depth map comes from SfM/sensors (ORB, RMSE=0.247

vs 0.427 for G2-MD [57]; LiDAR-8, RMSE=0.231 vs 0.306 for G2-MD [57]). These results show the superior robustness of our model across various densities, noise levels, and sensor types.

4.7. Ablation Studies

We randomly pick an indoor and an outdoor scene from the ETH3D-SfM dataset for validation (no overlap with the test set). We also sample 1000 images from the KITTI IDC training set for ablation studies (as we don't use it for training).

For the DDI ablation, we use the full training schedule, as the effectiveness is most obvious when models fully converge. Models for all other experiments are trained on 1/10 amount of the full data due to resource constraints.

Results are shown in Tab. 4. 1) *Multi-res DDI*: the improvement is most obvious on ETH3D, where the sparse depth maps contain large holes. When using 3 resolutions, the RMSE reduces to 0.459 from 0.595 for the vanilla DDI.

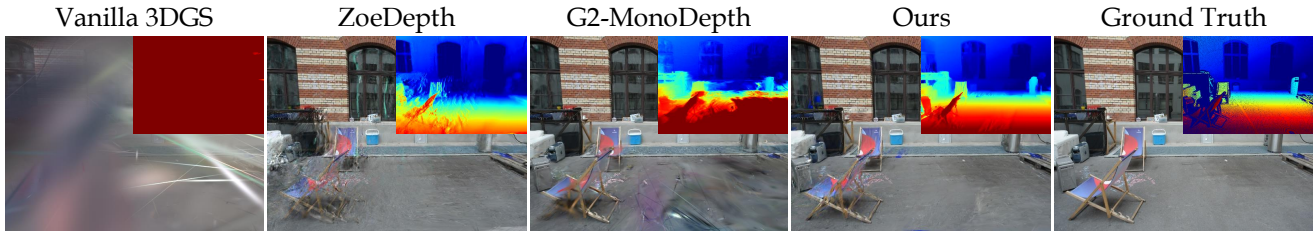


Figure 5. Rendered images and depths on test views. We train 3DGS [22] with a depth loss against depth predicted by different models.

Table 4. Ablation studies on the validation set. Res=1 is the vanilla DDI in OGNI-DC. ‘‘Synthetic’’ means the SfM+LiDAR patterns.

Methods		ETH3D-SfM		KITTI-64	
		RMSE	REL	RMSE	MAE
Multi-res DDI	DDI, Res=1	0.595	0.086	1.210	0.275
	DDI, Res=1,2	0.489	0.069	1.218	0.275
	DDI, Res=1,2,3	0.459	0.064	1.188	0.267
Losses	L_1	0.666	0.080	1.234	0.280
	L_1+L_{Lap}	0.598	0.083	1.224	0.278
	L_1+L_{gm}	0.547	0.082	1.155	0.282
	$L_1+L_{Lap}+L_{gm}$	0.490	0.076	1.173	0.277
Depth Space	Linear	0.886	0.155	1.289	0.305
	Log	0.627	0.103	1.293	0.308
	Log+Normalize	0.490	0.076	1.173	0.278
Training Pattern	Random	0.714	0.117	1.490	0.342
	Rand.+Synthetic	0.647	0.089	1.402	0.336
	Rand.+Syn.+Noise	0.490	0.076	1.173	0.278

Table 5. Results on the VOID [64] dataset under three densities.

Methods	VOID-1500		VOID-500		VOID-150	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Depth Pro [6]	0.734	0.385	0.697	0.373	0.758	0.392
DA-v2 [70]	0.605	0.209	0.582	0.209	0.644	0.230
Marigold [21]	0.630	0.240	0.607	0.241	0.673	0.263
CFormer [73]	0.726	0.261	0.821	0.385	0.956	0.487
DFU [62]	3.222	2.297	3.628	2.648	4.521	3.356
BP-Net [50]	0.738	0.268	0.790	0.369	0.934	0.470
DPrompting [39]	0.779	0.373	0.754	0.373	0.820	0.398
OGNI-DC [74]	0.593	0.175	0.589	0.198	0.693	0.261
G2-MD [57]	0.568	0.159	0.574	0.182	0.691	0.247
Ours	0.555	0.150	0.551	0.164	0.650	0.211

The metrics on KITTI are also improved. 2) *Losses*: while both the Laplacian loss and the gradient-matching loss lead to improvements over L_1 alone, combining them yields the best performance. 3) *Depth Normalization*: using the log-depth alone sacrifices the accuracy on KITTI, as the space on the numerical axis for larger depth values is compressed. Using the log-depth plus our normalization leads to improvements on both datasets. 4) *Training Depth Patterns*: using the synthetic patterns (SIFT, LiDAR) is better than training with only random samples, and injecting noise during training further boosts the performance.

4.8. Application: Novel View Synthesis

We show a practical application of OMNI-DC on view synthesis. We run OMNI-DC on the sparse depth map from

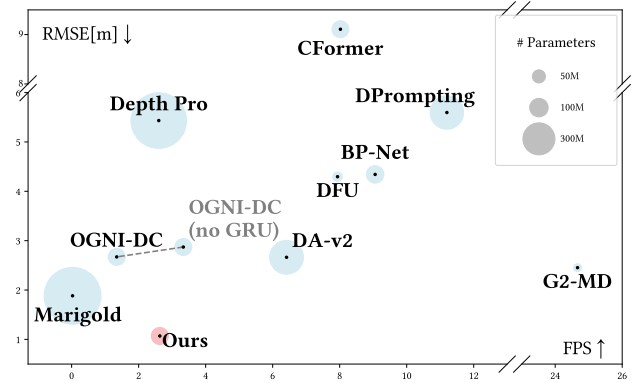


Figure 6. All numbers are benchmarked on the ETH3D [47] outdoor split with SfM points. We use a single 3090 GPU and an image resolution of 480×640 . Our model achieves the best accuracy with a small model size (85M vs 907M for Depth Pro) and a competitive running speed ($93 \times$ faster than Marigold [21] and $2 \times$ faster than OGNI-DC [74]). As an ablation, we report the speed of OGNI-DC without its iterative GRU updates. Our multi-resolution design brings slightly higher latency (300ms vs 379ms, +26.6%) compared to ‘‘OGNI-DC (no GRU)’’, but much better accuracy.

COLMAP, and follow DN-Splatter [54] to regularize the 3DGS with an additional depth loss. We test on the 13 scenes from ETH3D, and split 1/8 of the views for validation following the standard practice. The large scale of the scenes and the small overlaps among views make it a challenging dataset for view synthesis. Further details are provided in Appendix Sec. E.

Table 6. The novel view synthesis metrics and the rendered depth accuracy averaged on the 13 scenes from ETH3D.

Methods	3DGS	ZoeDepth [4]	G2-MD [57]	Ours
PSNR \uparrow	15.64	18.96	19.36	20.38
SSIM [63] \uparrow	0.557	0.573	0.641	0.660
LPIPS [72] \downarrow	0.418	0.324	0.273	0.229
RMSE (Depth) \downarrow	3.857	2.163	1.904	0.838

Results are shown in Tab. 6 and Fig. 5. The rendering and depth quality greatly improves compared to raw 3DGS, or using ZoeDepth or G2-MD for depth supervision.

5. Conclusion

We have introduced OMNI-DC, a highly robust depth completion model that performs well on different datasets and sparse patterns, with novel designs spanning model architecture, loss, and training data. We hope OMNI-DC will serve as a plug-and-play model for downstream tasks.

Acknowledgments

This work was partially supported by the National Science Foundation. We thank Jing Wen and all Princeton Vision & Learning Lab members for their insightful discussions and detailed comments on the manuscript.

References

- [1] Olivier Aubreton, Alban Bajard, Benjamin Verney, and Frederic Truchetet. Infrared system for 3d scanning of metallic surfaces. *Machine vision and applications*, 24:1513–1524, 2013. 4
- [2] 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, 12
- [3] 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. 6, 16
- [4] Shariq Farooq Bhat, Reiner Birkel, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 2, 3, 8, 13
- [5] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *ICML*, pages 1613–1622, 2015. 5
- [6] 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. 3, 6, 7, 8, 16, 19, 23, 24, 25
- [7] 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. 13
- [8] Manuel Carranza-García, F Javier Galán-Sales, José María Luna-Romera, and José C Riquelme. Object detection using depth completion and camera-lidar fusion for autonomous driving. *Integrated Computer-Aided Engineering*, 29(3): 241–258, 2022. 1
- [9] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE TPAMI*, 2019. 2
- [10] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *AAAI*, 2020. 2
- [11] 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. 13
- [12] Andrea Conti, Matteo Poggi, Filippo Aleotti, and Stefano Mattoccia. Unsupervised confidence for lidar depth maps and applications. In *IROS*, pages 8352–8359, 2022. 6
- [13] Andrea Conti, Matteo Poggi, and Stefano Mattoccia. Sparsity agnostic depth completion. In *WACV*, pages 5871–5880, 2023. 1, 2, 12
- [14] 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. 13
- [15] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *ECCV*, pages 241–258, 2024. 3
- [16] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, pages 4340–4349, 2016. 6
- [17] Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, and Marc Pollefeys. 3d visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68:14–27, 2017. 1
- [18] 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. 14
- [19] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *CVPR*, pages 2583–2592, 2021. 6, 16
- [20] HyunJun Jung, Patrick Ruhkamp, Guangyao Zhai, Nikolas Brasch, Yitong Li, Yannick Verdie, Jifei Song, Yiren Zhou, Anil Armagan, Slobodan Ilic, et al. On the importance of accurate geometry data for dense 3d vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 780–791, 2023. 4
- [21] 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. 2, 3, 6, 7, 8, 16, 19, 23, 24, 25
- [22] 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. 1, 2, 8, 13
- [23] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 15
- [24] 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. 6, 12, 16
- [25] 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. 12, 13
- [26] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with

- residual log-likelihood estimation. In *ICCV*, pages 11025–11034, 2021. 5
- [27] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. 5
- [28] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *AAAI*, pages 1638–1646, 2022. 1, 2, 3, 6, 14
- [29] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *NeurIPS*, 2017. 2
- [30] Xin Liu, Xiaofei Shao, Bo Wang, Yali Li, and Shengjin Wang. Graphcspn: Geometry-aware depth completion via dynamic gcn. In *ECCV*, 2022. 2
- [31] 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. 12
- [32] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision (IJCV)*, 60:91–110, 2004. 6, 7, 15, 23, 24
- [33] 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. 13
- [34] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012. 1, 2, 6
- [35] 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. 15
- [36] Hyungseob Park, Anjali Gupta, and Alex Wong. Test-time adaptation for depth completion. In *CVPR*, pages 20519–20529, 2024. 3, 12
- [37] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *ECCV*, pages 120–136, 2020. 1, 2
- [38] 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. 3, 12
- [39] Jin-Hwi Park, Chanhwi Jeong, Junoh Lee, and Hae-Gon Jeon. Depth prompting for sensor-agnostic depth estimation. In *CVPR*, pages 9859–9869, 2024. 2, 6, 7, 8, 12, 16, 19, 25
- [40] Jiexiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *CVPR*, pages 3313–3322, 2019. 6
- [41] 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. 2, 3, 5, 15
- [42] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, pages 10912–10922, 2021. 6
- [43] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *CVPR*, pages 12892–12901, 2022. 1, 5
- [44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 14
- [45] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, pages 2564–2571, 2011. 6, 7, 23, 24
- [46] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 5, 16
- [47] 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, 6, 8, 13, 16
- [48] 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. 12
- [49] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021. 5
- [50] Jie Tang, Fei-Peng Tian, Boshi An, Jian Li, and Ping Tan. Bilateral propagation network for depth completion. In *CVPR*, pages 9763–9772, 2024. 1, 2, 6, 7, 8, 16, 19, 23, 24, 25
- [51] Yifu Tao, Marija Popović, Yiduo Wang, Sundara Tejaswi Digumarti, Nived Chebroolu, and Maurice Fallon. 3d lidar reconstruction with probabilistic depth completion for robotic navigation. In *IROS*, pages 5339–5346, 2022. 1
- [52] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020. 6
- [53] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. Pdc-net+: Enhanced probabilistic dense correspondence network. *IEEE TPAMI*, 45(8):10247–10266, 2023. 5
- [54] 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. 1, 2, 8, 13
- [55] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017. 1, 2, 6, 16
- [56] 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. 6, 16

- [57] Haotian Wang, Meng Yang, and Nanning Zheng. G2-monodepth: A general framework of generalized depth inference from monocular rgb+ x data. *IEEE TPAMI*, 2023. [2](#), [3](#), [6](#), [7](#), [8](#), [12](#), [13](#), [16](#), [19](#), [23](#), [24](#), [25](#)
- [58] 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. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021. [6](#)
- [59] 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 *IROS*, 2020. [6](#)
- [60] 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. [1](#), [2](#), [15](#)
- [61] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *ECCV*, pages 36–54, 2024. [5](#), [14](#)
- [62] 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. [1](#), [2](#), [6](#), [7](#), [8](#), [16](#), [19](#), [23](#), [24](#), [25](#)
- [63] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. [8](#)
- [64] Alex Wong, Xiaohan Fei, Stephanie Tsuei, and Stefano Soatto. Unsupervised depth completion from visual inertial odometry. *IEEE Robotics and Automation Letters*, 5(2): 1899–1906, 2020. [2](#), [6](#), [8](#)
- [65] Guangkai Xu, Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Simon Chen, and Jia-Wang Bian. Towards domain-agnostic depth completion. *Machine Intelligence Research*, 21(4):652–669, 2024. [2](#)
- [66] Zheyuan Xu, Hongche Yin, and Jian Yao. Deformable spatial propagation networks for depth completion. In *ICIP*, 2020. [2](#)
- [67] 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. [12](#), [13](#)
- [68] 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. [12](#), [13](#)
- [69] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, pages 10371–10381, 2024. [3](#), [5](#)
- [70] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *NeurIPS*, 37:21875–21911, 2024. [2](#), [3](#), [5](#), [6](#), [7](#), [8](#), [12](#), [13](#), [16](#), [17](#), [19](#), [23](#), [24](#), [25](#)
- [71] 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. *CVPR*, 2020. [6](#)
- [72] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. [8](#)
- [73] 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](#), [5](#), [6](#), [7](#), [8](#), [12](#), [13](#), [14](#), [15](#), [16](#), [19](#), [23](#), [24](#), [25](#)
- [74] 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](#), [6](#), [7](#), [8](#), [12](#), [13](#), [15](#), [16](#), [19](#), [23](#), [24](#), [25](#)

OMNI-DC: Highly Robust Depth Completion with Multiresolution Depth Integration

Appendix

A. Comparison with Additional Generalizable DC Baselines

The codes are not available for some of the generalizable DC baselines [13, 38], 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 [39]	Fully Supervised	0.105	1.086	1.642
SpAgNet [13]		0.114	0.845	2.691
VPP4DC [2]		0.117	0.099	-
TTADC [36]	Test-Time Adapt.	0.204	-	-
UniDC [38]	100-Shot	0.147	1.224	2.890
Drompting [39]		0.175	1.275	4.587
UniDC [38]	Zero-Shot	0.323	4.061	-
VPP4DC [2]		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 [74] and CompletionFormer [73]).

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 OGNIDC and by 36.1% from CFormer on KITTI).

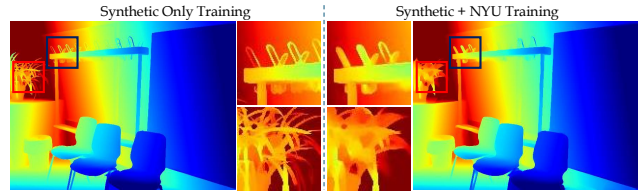


Figure a. Mixing NYU for training produces blurry depth maps on iBims [24].

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 [25]. 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 [25] 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 [31]	4129.7	31.853	4625.2	31.877	4760.0	31.879
Singh <i>et al.</i> [48]	3704.6	35.342	4137.1	35.166	4309.3	35.133
RadarCam [25]	2817.4	22.936	3117.7	22.853	3229.0	22.838
DA-v2 [70]	5466.5	47.446	6261.3	47.118	6566.9	47.053
OGNI-DC [74]	7612.7	29107.2	8151.2	28800.5	8356.9	28739.0
G2-MD [57]	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

D. Robustness to Night Time and Bad Weather

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

Carla-Night-DC [67] contains night-time driving scenes. LDCNet [67] 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) [68] dataset consists of real images from driving scenes captured at different weathers.

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.

Training	In-Domain	Zero-Shot, Indoors				Zero-Shot, Outdoors			
	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 [74], Synthetic Only	0.125	0.164	0.124	0.063	0.024	0.573	0.333	2.332	1.846
OGNI-DC [74], Synthetic + NYU	0.120	0.166	0.127	0.064	0.023	0.595	0.337	2.411	1.850
CFormer [73], Synthetic Only	0.130	0.176	0.148	0.064	0.030	0.595	0.359	2.338	2.037
CFormer [73], 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 [67]				DS-Sunny [68]		DS-Rainy [68]		DS-Foggy [68]		DS-Cloudy [68]	
	RMSE↓	MAE↓	iRMSE↓	iMAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓
LDCNet [67]	7.214	2.014	0.0546	0.0156	-	-	-	-	-	-	-	-
DA-v2 [70]	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 [74]	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 [57]	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

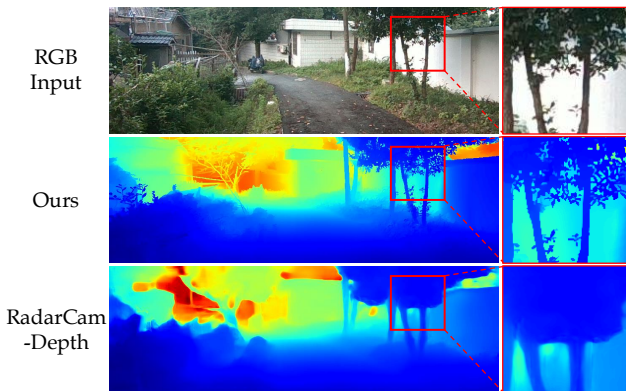


Figure b. Results on the Radar depth completion task. Our depth maps are much sharper than RadarCam-Depth [25].

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 [33] or 3DGS [22] 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 [11, 14]. We follow the recent work DN-Splatter [54], and use a depth

loss to train 3DGS. The loss can be written as:

$$\mathcal{L} = \mathcal{L}_{\hat{C}} + 0.2 \cdot \mathcal{L}_{\hat{D}}, \quad (13)$$

where $\mathcal{L}_{\hat{C}}$ is the original photometric loss in 3DGS [22], and $\mathcal{L}_{\hat{D}}$ is the edge-aware depth loss proposed in [54].

We evaluate on the ETH3D [47] 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 [4], and the depth completion model G2-MD [57]. For ZoeDepth, we align the scale and shift against the COLMAP sparse depth, following DN-Splatter [54]. 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, MVFormer++ [7].

We randomly split 1/8 of the view as test views and use the rest for training. The training follows the [54] 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.

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.

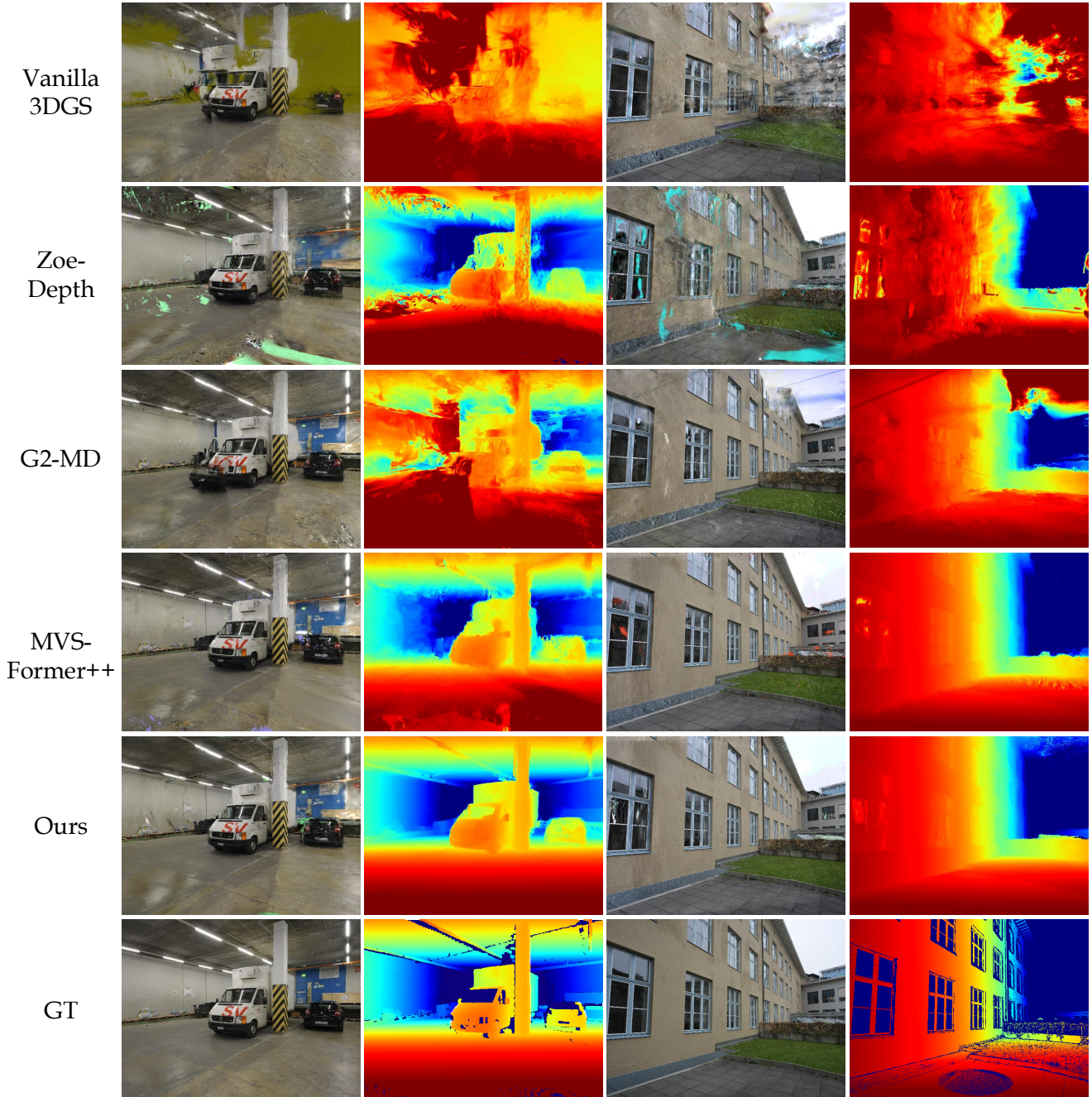


Figure c. Visualization of the rendered images and rendered depth maps against ground-truth on test views of the ETH3D dataset. The vanilla 3DGS is trained with only the photometric loss, and all other rows are trained with a depth loss against the predicted depth maps of the corresponding models. Our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

F. Implementation Details

F.1. Model Architecture and Loss Functions

We use the CompletionFormer [73] as the backbone. CompletionFormer is a U-Net-like [44] architecture with a feature pyramid. We extract the depth gradients by using the 1/4 resolution feature map with a series of ResNet [18] blocks and MaxPool2D layers, to obtain the depth gradi-

ents at the 1/4, 1/8, and 1/16 resolution.

From the full-resolution feature map, we extract the parameters for the DySPN [28] (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 [61], and predict γ from a Conv layer. We clamp the minimum value of γ to -2.0 to stabilize training.

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 \uparrow	15.64	18.96	19.36	20.02	20.38
SSIM \uparrow	0.557	0.573	0.641	0.644	0.660
LPIPS \downarrow	0.418	0.324	0.273	0.254	0.229
RMSE (Depth) \downarrow	3.857	2.163	1.904	1.847	0.838

To better deal with the noise in the input depth, we follow OGNI-DC [74] and use a 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):

$$\mathcal{E}_O = \sum_{i,j}^{W,H} M_{i,j} \cdot C_{i,j} \cdot (D_{i,j} - O_{i,j})^2 \quad (14)$$

When $C_{i,j} \rightarrow 0$, 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 auxiliary binary cross-entropy loss to directly supervise the confidence map.

The gradient-matching loss is implemented following MegaDepth [41] and MiDaS [41]:

$$\mathcal{L}_{\text{gm}} = \frac{1}{HW} \sum_{k=1}^4 \sum_{i,j}^{W,H} (|\nabla_x R_{i,j}^k| + |\nabla_y R_{i,j}^k|), \quad (15)$$

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 [23] optimizer with an initial learning rate of $1e-3$, for a total of 72 epochs. The learning rate decays by half at the 36th, 48th, 56th, and 64th epochs, following [73].

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 yields 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 [32] keypoints. For the random and SfM points, we also inject 0% \sim 5%

noisy depths by random sampling between the 5th and 95th 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 [60] 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 [35]. 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 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.

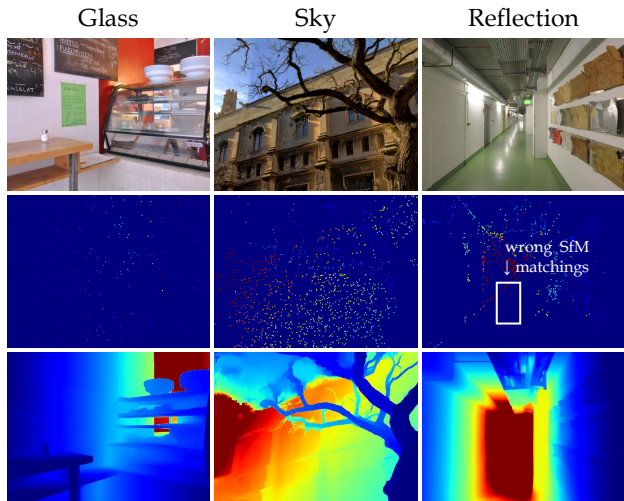


Figure d. Failure cases of OMNI-DC. Our model makes erroneous predictions when the scene contains glasses or reflective surfaces, as the depth sensor or multiview matching may fail. The sky cannot be naturally represented in the linear depth space.

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 [62] on NYU-500). On NYU-5, our method works better than all DC baselines (RMSE=0.536 vs 0.633 for OGNI-DC [74]).

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 [73]	0.090	0.012	0.141	0.021	0.429	0.092	0.707	0.181	1.141	0.307
	DFU [62]	0.091	0.011	-	-	-	-	-	-	-	-
	BP-Net [50]	0.089	0.012	0.132	0.021	0.414	0.090	0.609	0.157	0.869	0.294
	DPrompting [39]	0.105	0.015	0.144	0.023	0.178	0.031	0.213	0.043	0.380	0.095
	OGNI-DC [74]	0.089	0.012	0.124	0.018	0.157	0.025	0.207	0.038	0.633	0.171
Zero-shot	Depth Pro [6]	0.266	0.062	0.266	0.062	0.266	0.062	0.266	0.062	0.266	0.062
	DA-v2 [70]	0.309	0.061	0.309	0.061	0.314	0.062	0.330	0.063	0.814	0.136
	Marigold [21]	0.426	0.115	0.428	0.116	0.431	0.117	0.436	0.118	0.545	0.150
	G2-MD [57]	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

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 [62]). On the extremely sparse case (NYU-5), our method works better than OGNI-DC [74] and G2-MD [57], although worse than the monocular depth methods such as Depth Pro [6].

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 [70], OGNI-DC [74], and G2-MonoDepth [57]. 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 [24] consists of 100 indoor scenes captured with a laser scanner. The original images are at 480×640 resolution.

ARKitScenes [3] 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 [47]’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 [46] reconstruction for each scene into 2D to construct the sparse depth map.

DIODE [56]’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 [55]’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 [19]. 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

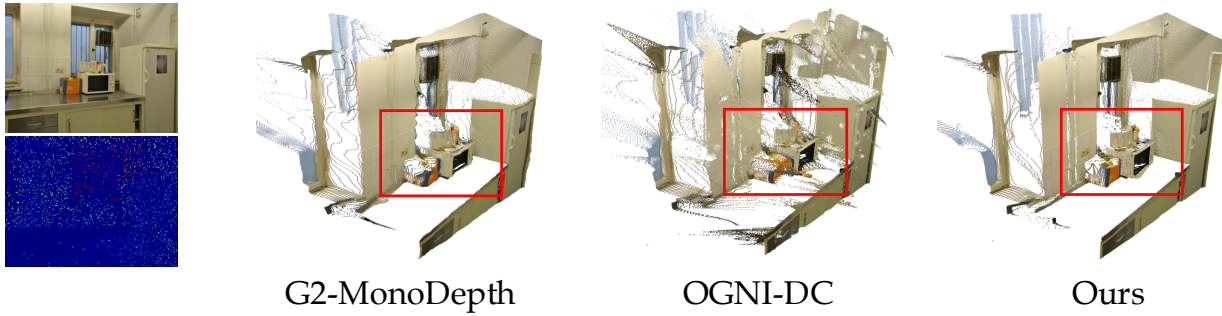
Most of the experiments in this paper are conducted under the resolution of 480×640 . However, modern cameras can often capture images at a higher resolution, which captures more details. Therefore, it is desirable that our DC model can work under higher resolutions.

We feed OMNI-DC with high-resolution images at test time. As shown in Tab. g, the inference time is $2.1 \times$ and $3.6 \times$ longer when tested on images with $2 \times$ and $2.7 \times$ resolution, respectively, a lower rate compared to the increase in pixel count. The memory consumption is 11.1GB when

ETH3D-Outdoor-COLMAP depth



iBims-0.7% density+10% Noise



KITTI-64-lines LiDAR

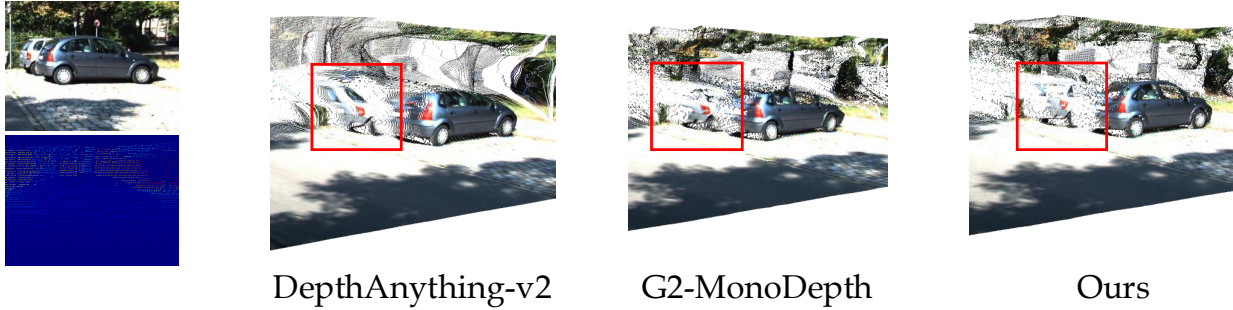


Figure e. The qualitative comparison of the 3D structures between our method and the best-performing baselines. On the outdoor scene from ETH3D, DA-v2 [70] has trouble capturing the global structure, while OGNI-DC’s reconstruction has distorted local details. On the noisy sparse depth map on iBims, the 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.

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.

Table g. Speed and 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

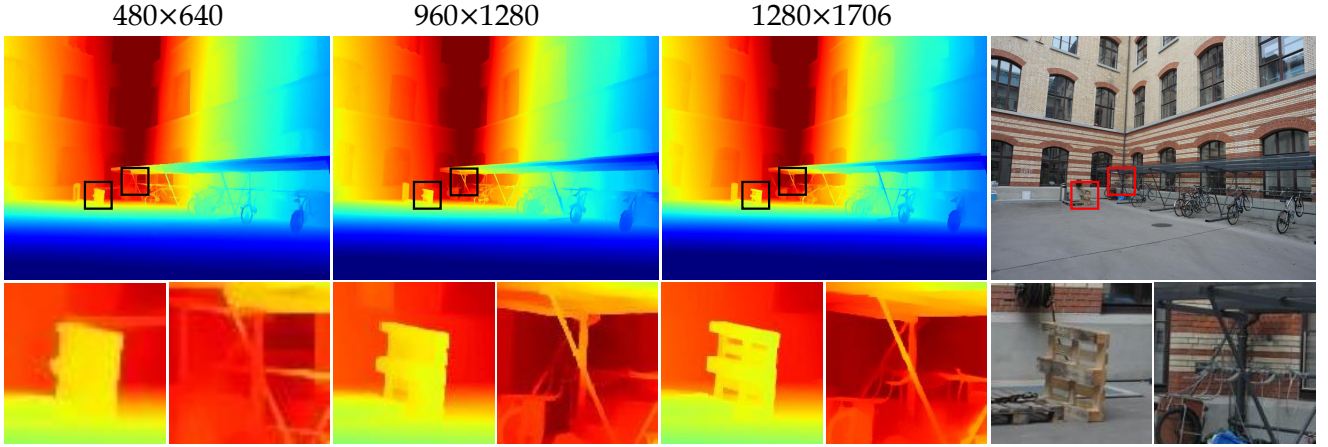


Figure f. More details are captured when running inference with higher resolution images at test time. All sparse depths are sampled under the 0.7% density.

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.*,

$$\hat{\mathbf{D}} = F(\mathbf{I}, \mathbf{O}). \quad (16)$$

We formally define the equivariance property as follows:

$$F(\mathbf{I}, \beta \cdot \mathbf{O}) = \beta \cdot F(\mathbf{I}, \mathbf{O}), \forall \beta \in \mathbb{R}_+, \quad (17)$$

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:

$$\hat{\mathbf{G}} = F(\mathbf{I}, \tilde{\mathbf{O}}; \theta), \tilde{\mathbf{O}} = \log(\mathbf{O}) - \log(\text{median}(\mathbf{O})). \quad (18)$$

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

$$\begin{aligned} \tilde{\mathbf{O}}(\beta \cdot \mathbf{O}) &= \log(\beta \cdot \mathbf{O}) - \log(\text{median}(\beta \cdot \mathbf{O})) \\ &= \log(\beta) + \log(\mathbf{O}) - \log(\beta) - \log(\text{median}(\mathbf{O})) \\ &= \tilde{\mathbf{O}}(\mathbf{O}), \forall \beta \in \mathbb{R}_+. \end{aligned} \quad (19)$$

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

$$\hat{\mathbf{G}}(\mathbf{I}, \beta \cdot \mathbf{O}) = \hat{\mathbf{G}}(\mathbf{I}, \mathbf{O}), \forall \beta \in \mathbb{R}_+. \quad (20)$$

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:

$$\hat{\mathbf{D}}^{\log} = \arg \min_{\mathbf{D}^{\log}} \left(\alpha \cdot \mathcal{E}_O(\mathbf{D}^{\log}, \mathbf{O}, \mathbf{M}) + \mathcal{E}_G(\mathbf{D}^{\log}, \hat{\mathbf{G}}) \right), \quad (21)$$

where

$$\begin{aligned} \mathcal{E}_O &:= \sum_{i,j}^{W,H} \mathbf{M}_{i,j} \cdot (\mathbf{D}_{i,j}^{\log} - \log(\mathbf{O}_{i,j}))^2, \\ \mathcal{E}_G &:= \sum_{r=1}^R \sum_{i,j}^{W,H} \left(\mathbf{G}_{i,j}^x - \hat{\mathbf{G}}_{i,j}^x \right)^2 + \left(\mathbf{G}_{i,j}^y - \hat{\mathbf{G}}_{i,j}^y \right)^2, \end{aligned} \quad (22)$$

with $\mathbf{G}_{i,j}^{r,x} := \mathbf{D}_{i,j}^r - \mathbf{D}_{i-1,j}^r$; $\mathbf{G}_{i,j}^{r,y} := \mathbf{D}_{i,j}^r - \mathbf{D}_{i,j-1}^r$ 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. (21), 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. (22). Plugging $\log \beta + \mathbf{D}^{\log}$ and $\beta \cdot \mathbf{O}$ into Eq. (22) gives the exact same energy as \mathbf{D}^{\log} and \mathbf{O} .

Given Lemma 1, we finally have

$$\begin{aligned} \hat{\mathbf{D}}(\hat{\mathbf{G}}, \beta \cdot \mathbf{O}, \mathbf{M}) &= \exp\left(\hat{\mathbf{D}}^{\log}(\hat{\mathbf{G}}, \beta \cdot \mathbf{O}, \mathbf{M})\right) \\ &= \exp\left(\log \beta + \hat{\mathbf{D}}^{\log}(\hat{\mathbf{G}}, \mathbf{O}, \mathbf{M})\right) \quad (23) \\ &= \beta \cdot \hat{\mathbf{D}}(\hat{\mathbf{G}}, \mathbf{O}, \mathbf{M}), \forall \beta \in \mathbb{R}_+. \quad \square \end{aligned}$$

L.2. Empirical Evidence

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

Depth Scale	0.001×	0.1×	1×	10×	1000×
CFormer [73]	810.8	5.404	0.236	0.684	0.997
OGNI-DC [74]	7.079	0.704	0.158	0.387	0.622
G2-MD [57]	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:

$$\text{REL}(\hat{\mathbf{D}}, \mathbf{D}^{\text{gt}}) = \frac{1}{HW} \cdot \sum_{i,j} \frac{|\hat{\mathbf{D}}_{i,j} - \mathbf{D}_{i,j}^{\text{gt}}|}{\mathbf{D}_{i,j}^{\text{gt}}} \quad (24)$$

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 [6] 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 [21] (in linear depth space) and DepthAnythingv2 [70] (in disparity space).

For BP-Net [50], Depth Prompting [39], and OGNI-DC [74], we use their model trained on NYUv2 and KITTI for indoor and outdoor testing, respectively. We use the DFU [62] checkpoint trained on KITTI for all experiments, since its NYU code is not released. G2-MD [57] 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:

$$\begin{aligned} \text{MAE}(\hat{\mathbf{D}}, \mathbf{D}^{\text{gt}}) &= \frac{1}{HW} \cdot \sum_{i,j} |\hat{\mathbf{D}}_{i,j} - \mathbf{D}_{i,j}^{\text{gt}}| \\ \text{REL}(\hat{\mathbf{D}}, \mathbf{D}^{\text{gt}}) &= \frac{1}{HW} \cdot \sum_{i,j} \frac{|\hat{\mathbf{D}}_{i,j} - \mathbf{D}_{i,j}^{\text{gt}}|}{\mathbf{D}_{i,j}^{\text{gt}}} \end{aligned}$$

$$\begin{aligned} \text{RMSE}(\hat{\mathbf{D}}, \mathbf{D}^{\text{gt}}) &= \sqrt{\frac{1}{HW} \cdot \sum_{i,j} (\hat{\mathbf{D}}_{i,j} - \mathbf{D}_{i,j}^{\text{gt}})^2} \\ \delta_1(\hat{\mathbf{D}}, \mathbf{D}^{\text{gt}}) &= \frac{1}{HW} \sum_{i,j} \mathbf{1} \left(\max \left(\frac{\hat{\mathbf{D}}_{i,j}}{\mathbf{D}_{i,j}^{\text{gt}}}, \frac{\mathbf{D}_{i,j}^{\text{gt}}}{\hat{\mathbf{D}}_{i,j}} \right) < 1.25 \right) \end{aligned}$$

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 [57] and OGNI-DC [74], our method generates much sharper results and is more robust to noise. While DA-v2 [70] produces sharp details, its global structure is always off, especially for outdoor scenes.

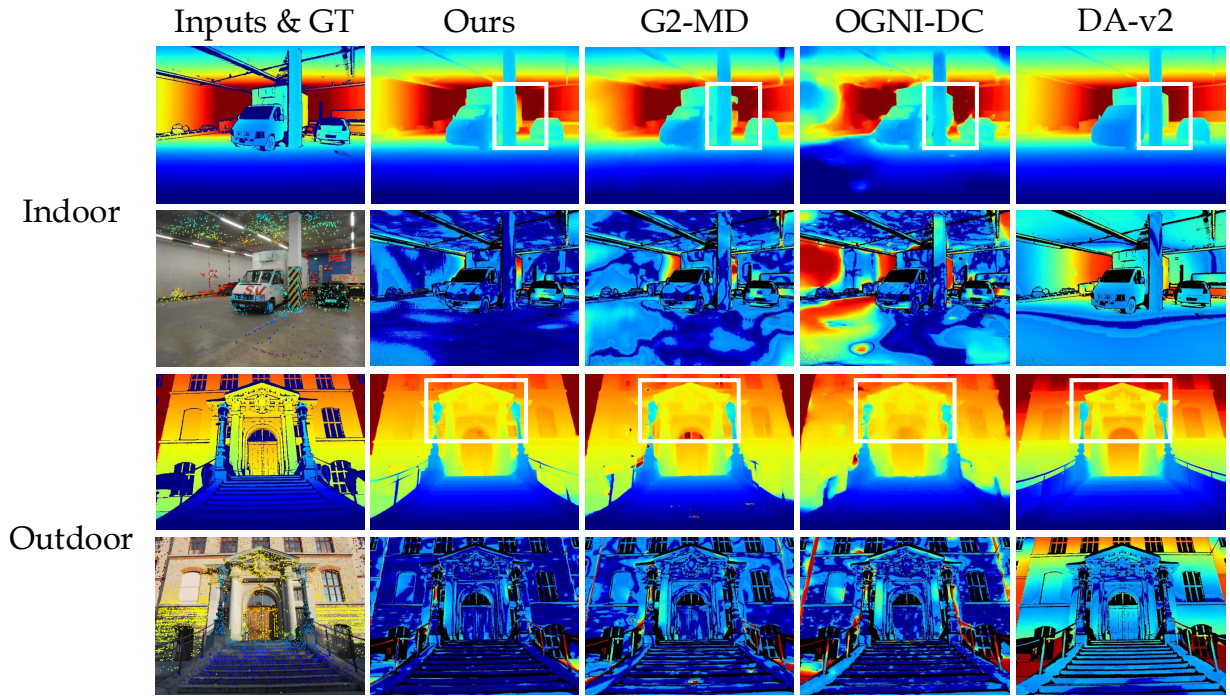
P. More Ablations on the Laplacian Loss

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.

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
$L_{Lap} + L_{gm} + L_1$	0.490	0.076	1.173	0.277

ETH3D-SfM



KITTI-LiDAR-64 Lines

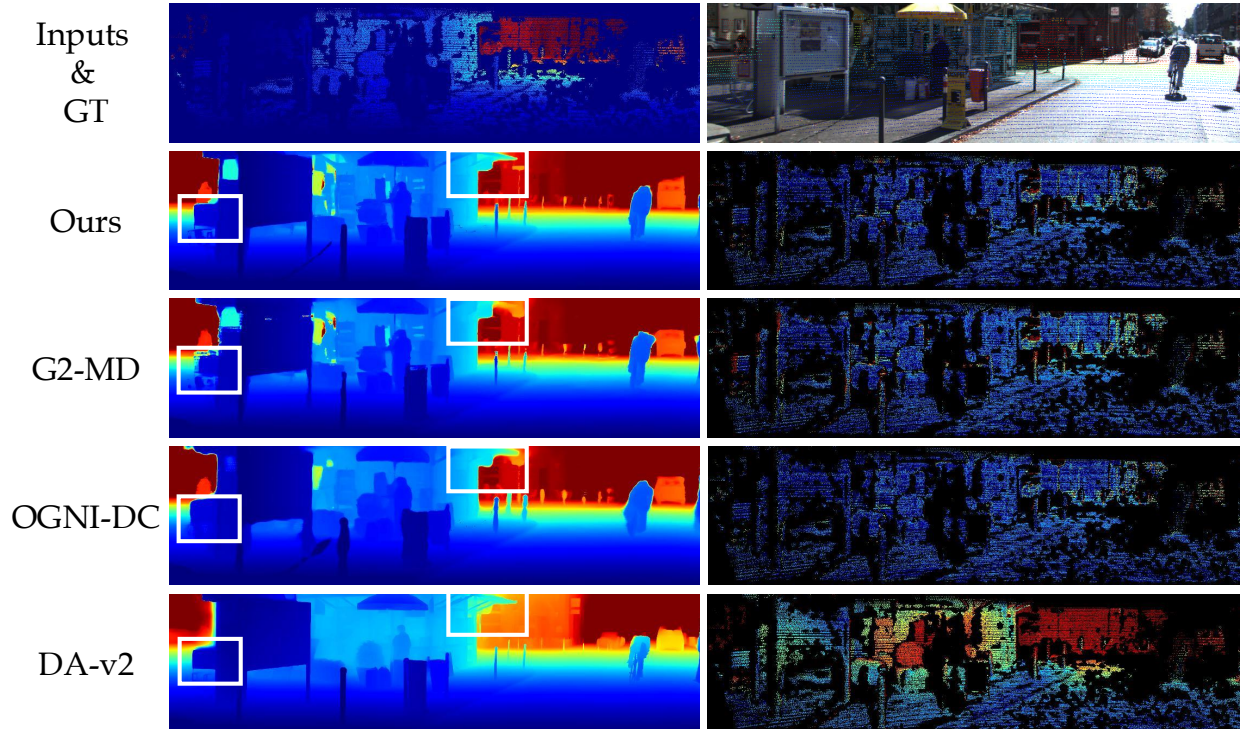


Figure g. First row/column: gt and predicted depth; second row/column: RGB, sparse depth (superimposed), and error maps (blue means small errors).

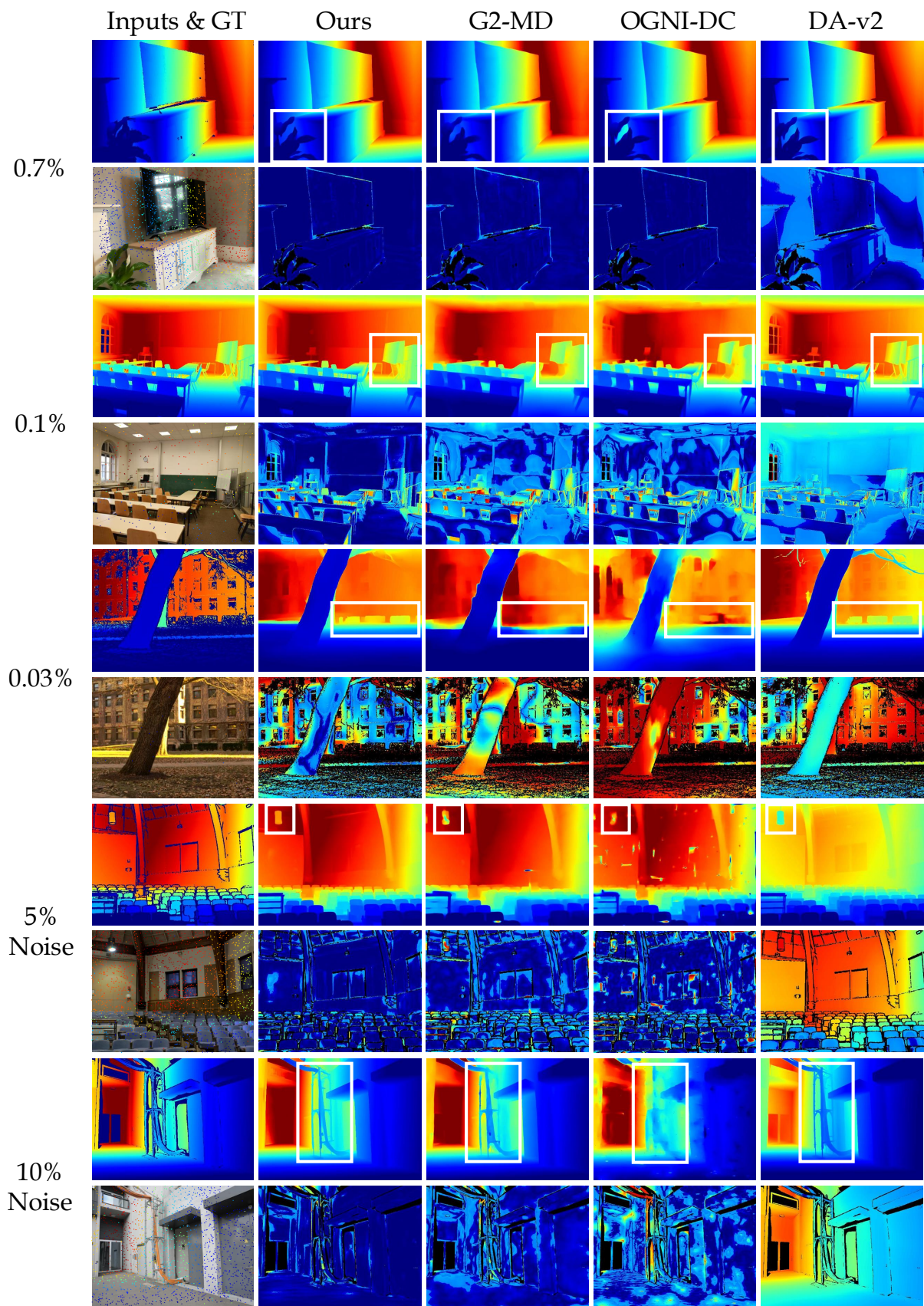


Figure h. First row: gt and predicted depth; second row: RGB, sparse depth (superimposed), and error maps (blue means small errors).

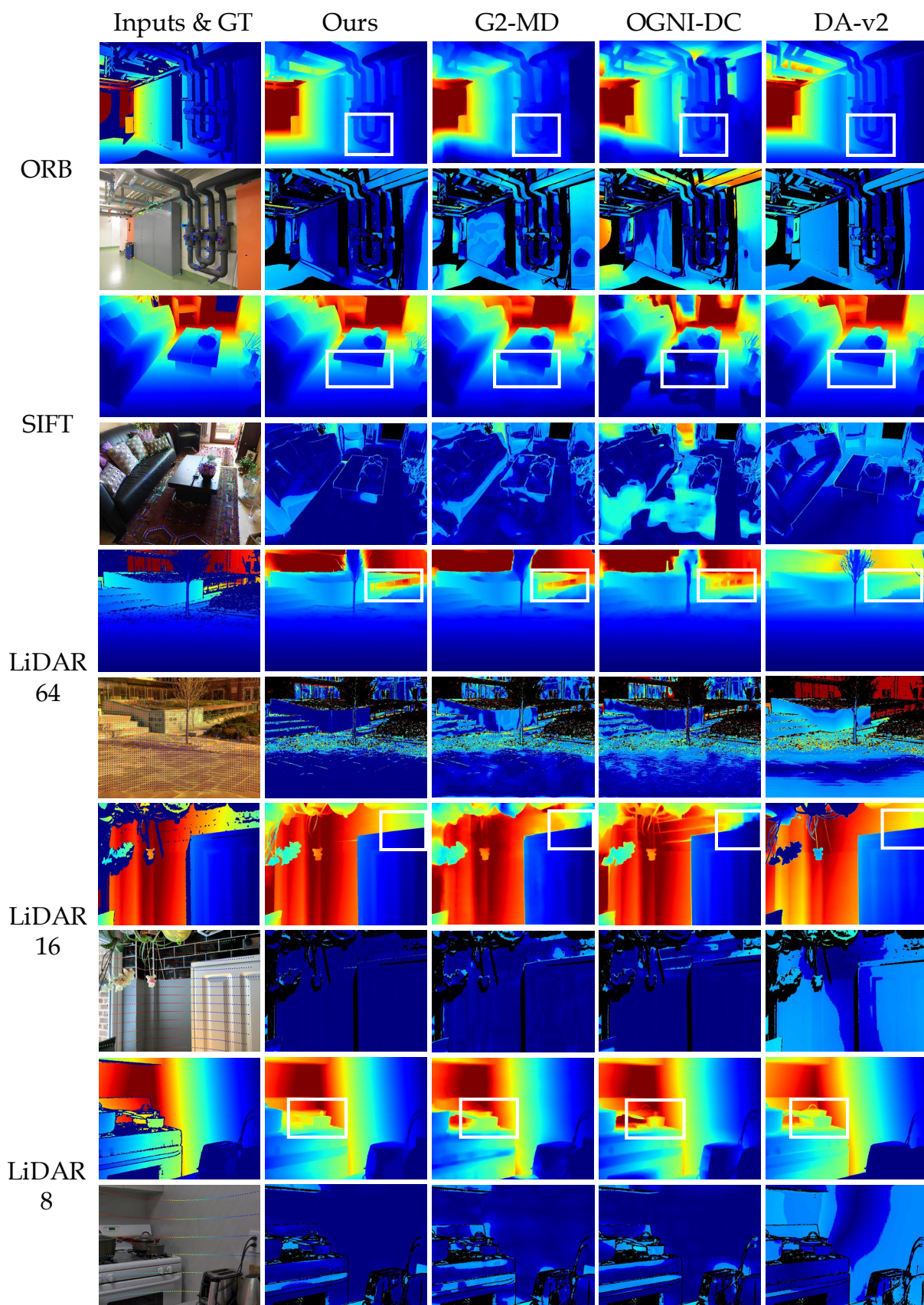


Figure i. First row: gt and predicted depth; second row: RGB, sparse depth (superimposed), and error maps (blue means small errors).

Table j. Quantitative comparison with baselines on the **synthetic depth patterns** on the **indoor scenes**. Results averaged on the ARK-itScenes, 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 [6]	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 [70]	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 [21]	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 [73]	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 [62]	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 [50]	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 [74]	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 [57]	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 [45]			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
Depth Pro [6]	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 [70]	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 [21]	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 [73]	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 [62]	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 [50]	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 [74]	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 [57]	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 [32]				LiDAR-64-Lines				LiDAR-16-Lines			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
Depth Pro [6]	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 [70]	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 [21]	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 [73]	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 [62]	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 [50]	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 [74]	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 [57]	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 [6]	0.636	0.524	0.176	0.746
DA-v2 [70]	0.602	0.194	0.042	0.982
Marigold [21]	0.309	0.187	0.062	0.951
CFormer [73]	0.934	0.609	0.168	0.662
DFU [62]	4.022	3.029	2.141	0.257
BP-Net [50]	0.816	0.587	0.179	0.652
OGNI-DC [74]	0.287	0.114	0.028	0.979
G2-MD [57]	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 [6]	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 [70]	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 [21]	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 [73]	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 [62]	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 [50]	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 [74]	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 [57]	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 [45]			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
Depth Pro [6]	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 [70]	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 [21]	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 [73]	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 [62]	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 [50]	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 [74]	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 [57]	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 [32]				LiDAR-64-Lines				LiDAR-16-Lines			
	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1	RMSE	MAE	REL	δ_1
Depth Pro [6]	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 [70]	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 [21]	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 [73]	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 [62]	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 [50]	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 [74]	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 [57]	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 [6]	7.712	6.368	0.426	0.183
DA-v2 [70]	6.304	2.056	0.119	0.922
Marigold [21]	2.578	1.382	0.124	0.883
CFormer [73]	7.759	5.549	1.071	0.472
DFU [62]	5.242	3.027	0.401	0.623
BP-Net [50]	5.859	3.282	0.226	0.776
OGNI-DC [74]	3.354	1.671	0.197	0.824
G2-MD [57]	2.404	0.918	0.078	0.936
Ours	2.096	0.715	0.048	0.961

Table 1. Quantitative comparison with baselines on the ETH3D-SfM and KITTI-DC. 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 [73]	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 [62]	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 [50]	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 [39]	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 [74]	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 [6]	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 [70]	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 [21]	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 [57]	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 [73]	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 [62]	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 [50]	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 [39]	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 [74]	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 [6]	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 [70]	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 [21]	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 [57]	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