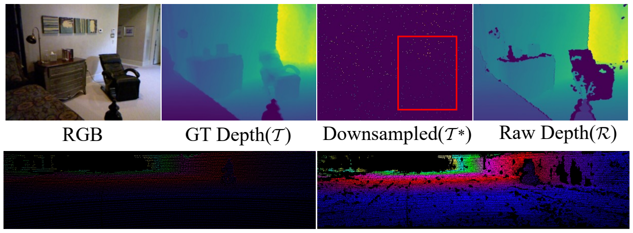
Supplementary Materials for RGB-Depth Fusion GAN for Indoor Depth Completion

1. Regular Downsampled Input vs. Raw input

The regular downsampled setting of most existing methods following Ma and Karaman [9] mimics well the task of outdoor depth completion from *raw Lidar scans* to *dense annotations*, as shown in the bottom of Fig. 1. However, for indoor RGB-depth sensor data, directly using downsampled input is improper: 1) The *raw depth* $\mathcal{R}$ captured by depth sensors is dense and continuous, which is quite different from the sparse pattern of *downsampled input* $\mathcal{T}^*$; 2) As shown in the red box in Fig. 1, the downsampled input reveals ground truth depth values to the models that can not be obtained in practice. Thus, we believe the raw input setting ($\mathcal{R} \Rightarrow \mathcal{T}$) is more practicable (and not only a specific case) for indoor depth completion than $\mathcal{T}^* \Rightarrow \mathcal{T}$.



RGB GT Depth($\mathcal{T}$) Downsampled($\mathcal{T}^*$) Raw Depth($\mathcal{R}$)

Raw Lidar Scans Dense Annotation

Figure 1. Depth data visualizations of indoor RGB-Depth sensors (top, NYU-Depth V2) and outdoor Lidar scans (bottom, KITTI).

2. More Details of Pseudo Depth Maps

Section 3.5 of the main paper introduces our proposed pseudo depth maps for training indoor depth completion methods. In this section, we provide more details about the design for the pseudo depth maps, including how and why the five masking methods are used for generating pseudo depth maps and a few more visualization results.

(1) *Highlight masking*. The RGB-D camera has difficulty in obtaining depth data of shiny surfaces because IR rays reflected from these surfaces are weak or scattered [6]. Meanwhile, these smooth and shiny objects often lead to specular highlights and bright spots in the RGB images. Hence, we detect these highlight regions in RGB images and mask them in depth maps to generate pseudo depth maps. We borrow from Arnold *et al*. [1] for highlight detection, which has a good balance of effectiveness and efficiency.

(2) *Black masking*. Since dark and matte surfaces are good absorbers and poor reflectors of radiation, the depth map is strongly affected by these surfaces [2]. We randomly mask the depth pixels whose values of R, G, and B in the RGB images are all in [0, 5], which can simply but directly handle some regions that are easy to have invalid depth values.

(3) *Graph-based segmentation masking*. The chaotic light reflections in the complex environment can interfere with the return of infrared light and cause discrete and irregular noises in depth maps. We use the graph-based segmentation [4] to divide the RGB image into several blocks of different sizes and mask the small blocks.

(4) *Semantic masking*. Some materials, such as glass, mirror, and porcelain surfaces, easily cause scattered infrared reflection and missing depth return values. We utilize the semantic label information to randomly cover objects probably containing these materials, such as TV, mirror, and window. We randomly mask all pixels for one or two objects in each frame.

(5) *Semantic XOR masking*. Similar motivations to the graph-based segmentation masking, we use semantic segmentation to recognize complex regions in the scene. We use the U-Net [13] network to randomly partition 20% of the training set for semantic segmentation task training and subsequently use it to semantically segment the remaining data. We take the regions where the predicted segmentation results are different from the ground-truth to be the complex regions, then mask the depth values in those regions.

Fig. 2 shows the quantitative results for each downsampling method. In Fig. 2(1), the highlight regions we masked are basically the depth missing regions of the raw depth images. We only randomly mask some sporadic black areas since the RGB image has a certain deviation from the real color, in Fig. 2(2). Graph-based segmentation masking simulates some discrete depth loss very well of depth maps in Fig. 2(3). In Fig. 2(4), semantic masking covers out some objects that may cause a lack of depth values. Semantic

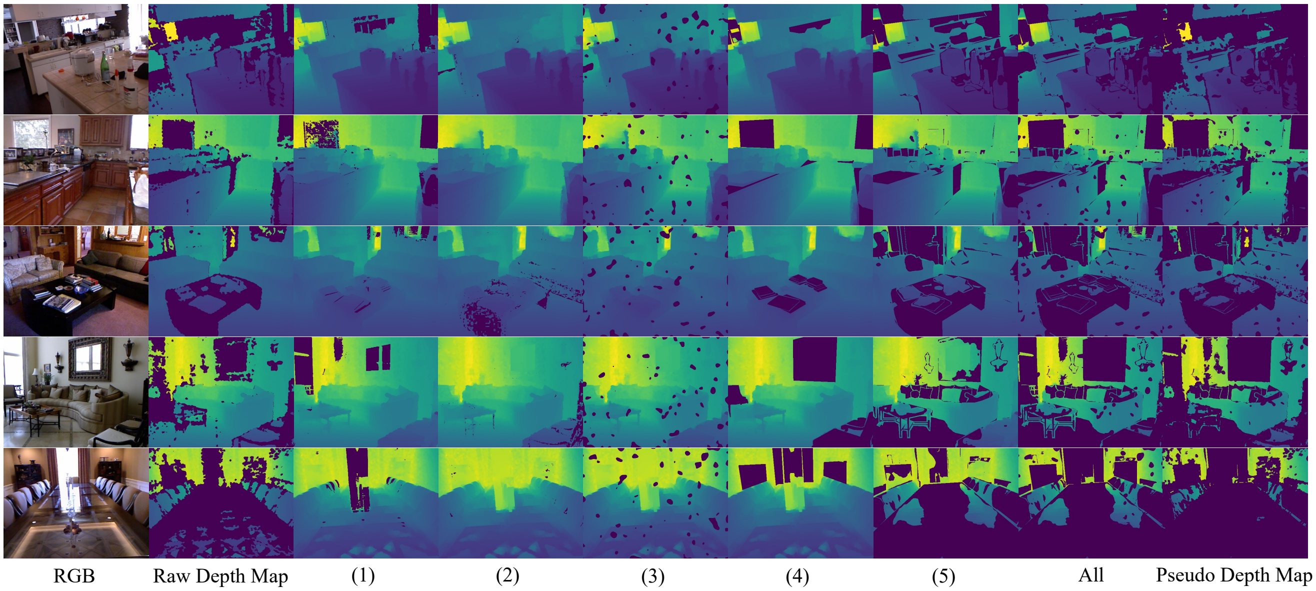


Figure 2. Visualizations of the five sampling methods. (1) Highlight masking. (2) Black masking. (3) Graph-based segmentation masking. (4) Semantic masking. (5) Semantic XOR masking. 'All' refers to the results of using all methods on the reconstruction of the depth map. 'Pseudo Depth Map' refers to the results of using all methods on the raw depth map.

XOR masking masks a wide range of regions where the predicted and ground-truth values differ in Fig. 2(5).

3. Three Training and Evaluation Settings

Setting	Training		Testing		
	$\mathcal{P} \Rightarrow \mathcal{R}$	$\mathcal{R}^* \Rightarrow \mathcal{R}$	$\mathcal{R} \Rightarrow \mathcal{T}$	$\mathcal{R}^* \Rightarrow \mathcal{T}$	$\mathcal{T}^* \Rightarrow \mathcal{T}$
A	✓		✓		
B		✓		✓	
C		✓			✓

Table 1. $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{P}$ represent the raw, reconstructed, and pseudo depth map, respectively. $\cdot^*$ represents the random sparse sampling with 500 valid depth pixels.

In the main paper, we set up three different test methods and corresponding training strategies, as shown in Tab. 1. $\mathcal{R}$ and $\mathcal{T}$ represent raw or incompleted depth images and reconstructed and completed depth maps, respectively. In the training set, due to the deficiency of a large number of reconstructed depth maps, most methods downsample the raw or incompleted depth images to predict the valid pixels of raw depth maps. In our work, we use the pseudo depth maps for training. In addition, we randomly sample 500 valid points $\mathcal{R}^*$ to get the sparse depth map as the input following existing methods [8,9]. The specific three evaluation programs are set up as follows:

- *Setting A*: At the training time, we use pseudo depth maps $\mathcal{P}$ as model input, and supervise with raw depth image. In testing, we input a raw depth map to predict the complemented and reconstructed depth map, which is most in line with the real scenario of indoor depth completion. Our method uses the pseudo depth maps, and other methods are trained in the synthetic semi-dense sensor data [14].
- *Setting B*: Although our model is not designed for sparse scenes, we use the sparse depth map $\mathcal{R}^*$ with randomly sampled 500 valid depth pixels following existing methods [3,8,9] for training to evaluate the model completion performance. At the test time, the input is consistent with the sampling method of training for raw depth images, and the reconstructed depth map is used as the ground truth for evaluation.
- *Setting C*: For comparing more existing methods [3,5,7–10,12] of depth completion, we randomly sample the 500 pixels in the reconstructed depth map as input at the test phase. This sampling method, despite the fact that only 500 valid points are the input, would have much better metrics than the above two sampling methods because of the accurate depth information obtained for all regions.

4. Object Detection after Depth Completion

We show extended experimental results using completed depth maps for 3D object detection, of which some representative results are shown in Section 4.4 in the main paper. We compare with the depth maps generated by DeepLidar [12] and NLSPN [10] on the 3D object detection task.

Method	mAP@25	mAP@50	RMSE
VoteNet [11]	59.07	35.77	-
DeepLidar [12] + VoteNet [11]	59.73	35.49	0.279
NLSPN [10] + VoteNet [11]	47.43	26.10	0.267
Ours + VoteNet [11]	**60.64**	**37.28**	**0.255**
H3DNet [15]	60.11	39.04	-
DeepLidar [12] + H3DNet [15]	60.35	39.16	0.279
NLSPN [10] + H3DNet [15]	27.10	9.77	0.267
Ours + H3DNet [15]	**61.03**	**39.71**	**0.255**

Table 2. Comparisons of 3D object detection results with the completed depth map on SUN RGB-D. The last column is the complementary result for DeepLidar, NLSPN, and Ours.

DeepLidar [12] uses a surface normal pathway to assist in depth map completion. NLSPN [10] learns the convolutional kernel size and iteration number for propagation to optimize the boundary depth. In Tab. 2, compared to DeepLidar [12], our model improves more significantly in all metrics. NLSPN [10] produces too much noise in the completion, which causes the performance of the detector to degrade.

References

[1] Mirko Arnold, Anarta Ghosh, Stefan Ameling, and Gerard Lacey. Automatic segmentation and inpainting of specular highlights for endoscopic imaging. *EURASIP Journal on Image and Video Processing*, 2010:1–12, 2010. 1

[2] Eu-Tteum Baek, Hyung-Jeong Yang, Soo-Hyung Kim, Gueesang Lee, and Hieyong Jeong. Distance error correction in time-of-flight cameras using asynchronous integration time. *Sensors*, 20(4):1156, 2020. 1

[3] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018. 2

[4] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision (IJCV)*, 59(2):167–181, 2004. 1

[5] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. Depth coefficients for depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[6] Sung-Yeol Kim, Manbae Kim, and Yo-Sung Ho. Depth image filter for mixed and noisy pixel removal in rgb-d camera systems. *IEEE Transactions on Consumer Electronics*, 59(3):681–689, 2013. 1

[7] Byeong-Uk Lee, Hae-Gon Jeon, Sunghoon Im, and In So Kweon. Depth completion with deep geometry and context guidance. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3281–3287. IEEE, 2019. 2

[8] Byeong-Uk Lee, Kyunghyun Lee, and In So Kweon. Depth completion using plane-residual representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13916–13925, June 2021. 2

[9] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4803. IEEE, 2018. 1, 2

[10] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *European Conference on Computer Vision (ECCV)*, pages 120–136. Springer, 2020. 2, 3

[11] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9277–9286, 2019. 3

[12] Jiaxiong 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3313–3322, 2019. 2, 3

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[14] Dmitry Senushkin, Mikhail Romanov, Ilia Belikov, Nikolay Patakin, and Anton Konushin. Decoder modulation for indoor depth completion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, pages 2181–2188. IEEE, 2021. 2

[15] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 311–329. Springer, 2020. 3