Flexible Depth Completion for Sparse and Varying Point Densities (Supplementary Material)

Jinhyung Park[1] Yu-Jhe Li[2] Kris Kitani[1]
[1] Carnegie Mellon University [2] Microsoft Research

A. Overview

In this supplementary, we provide additional dataset details, quantitative results, and qualitative visualizations. These sections are organized as follows:

- Section B provides additional details regarding the KITTI and NYUv2 depth completion datasets. Further, we provide additional details and visualization of our selected scan-lines on KITTI for our few-line setting.
- Section C contains additional implementation and training details of our Affinity-Based Shift Correction (ASC) module.
- Section D extends our ablation study in the main paper with tables containing additional metrics as well as with visualizations of intermediate depth maps.
- Sections E and F contain a more extended leaderboard comparison on the 64-line KITTI and 500-point NYUv2 settings.
- Section G presents a comparison of our method to Sparsity Agnostic Depth Completion [5].
- Section I contains additional details of the nuScenes depth completion dataset as well as visualizations for this difficult domain adaptation setting.
- Section J contains further discussion regarding our experiments using stronger monocular depth estimation models.
- Section K provides results of applying 3D detection method VoxelRCNN on the completed depth maps and demonstrates a single pipeline for 3D detection on variable sparsity LiDAR.

B. Additional Dataset Details

B.1. KITTI Dataset

The KITTI dataset [5, 12] contains 87k pairs of 64-line LiDAR depth maps and RGB images for training, 1,000 images for selected validation, and 1,000 images for online testing. We use the 1,000-image selected validation set in our experiments. With few LiDAR points at the top, the images are bottom-cropped to 256x1216 for training and testing similar to prior work. [23, 28, 35, 43, 45].

B.2. Scan Line Selection on KITTI Dataset

As the 64-line LiDAR in the KITTI dataset is quite dense, with each image pixel being within 5 pixels of a depth point, we evenly subsample the 64-line LiDAR to simulate more affordable fewer-line LiDAR sensors. To get scan lines, we transform points to spherical world coordinates and bin by zenith angle similar to prior work. In doing so, we find that depth completion performance varies greatly for 1, 2, 4, and 8 line LiDAR depending on the pitch of the simulated sensor. As the LiDAR sensor is carefully placed to ensure maximum scene coverage in realistic scenarios, we similarly carefully choose the best setup for each 1, 2, 4, and 8 line simulated sensor to maximize depth completion performance. As we are downsampling 64-line LiDAR to simulate fewer-line LiDAR, "pitch" of the simulated sensor is represented by choosing different line indicies (out of the 64 lines in the original KITTI sensor) for the first scan line of the fewer-line sensor. For instance, choosing index 0 for the 1-line LiDAR causes the single scan line to be pointed directly at the immediate ground location, resulting in poor depth completion performance. On the other hand, an index of 63 places the single scan line above most elements of the scene, similarly resulting in poor performance.

To ensure that the selected scan lines are not biased to any evaluated depth completion model, we use a separate model to determine the most suitable scan lines. More specifically, we use a separate model with the same standard architecture as MIDAS [31], a commonly used monocular model, but with RGBD input and a ResNet50 backbone. Note that while this model shares the same architecture as MIDAS, the pre-trained MIDAS weights are not used, and the backbone only has ImageNet pre-training. A model is trained for each # of scan lines over varying pitches.

The results of this model evaluated on various pitches is shown with different metrics in Figure 5. We find that depth completion performance varies greatly over different pitches for few-line sensors. Notably, we find that a poorly placed 4-line sensor (2542 RMSE at starting index 15) can even be outperformed by a optimally placed 2-line sensor (2414 RMSE at starting index 22), demonstrating that the

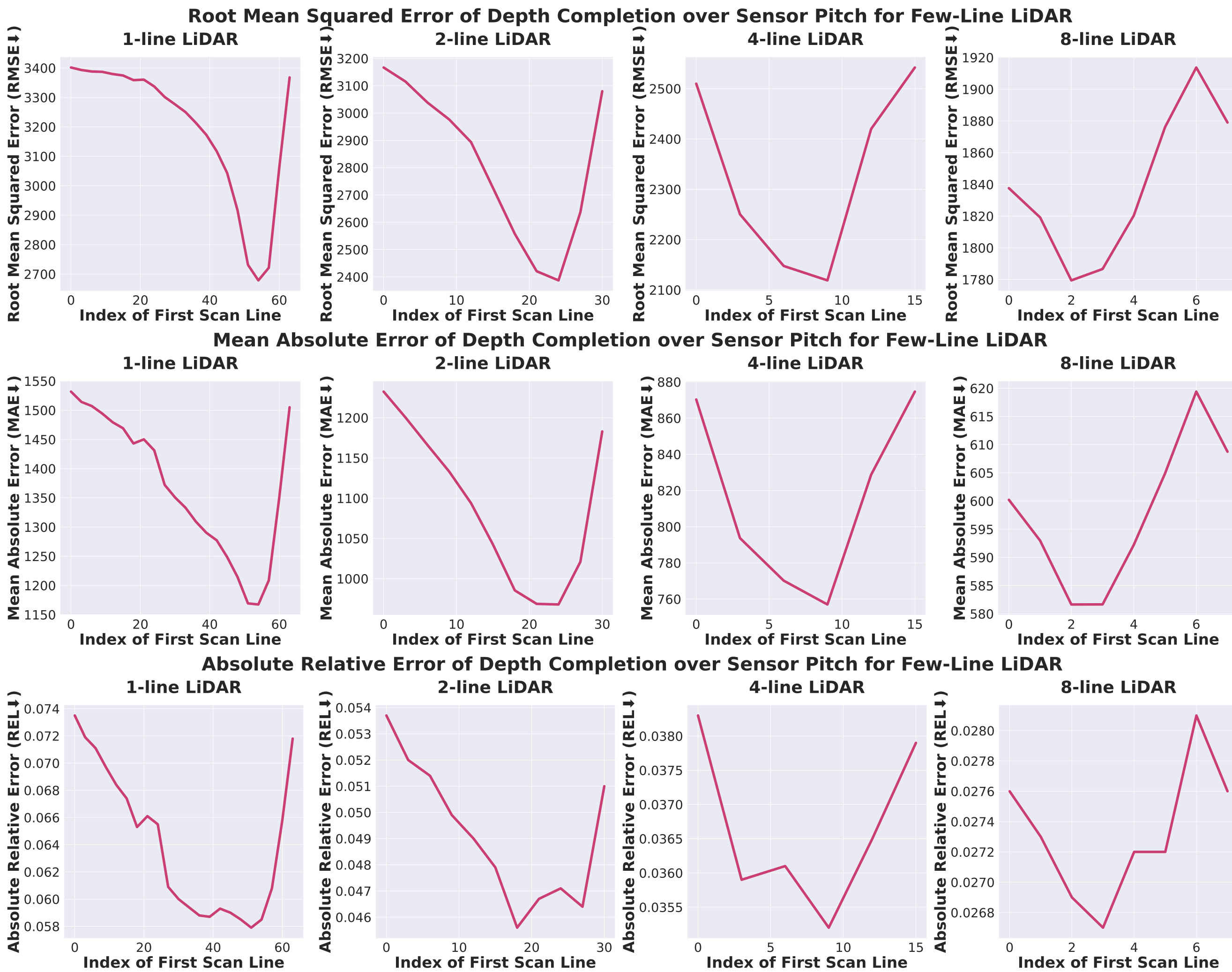

Figure 5. We show depth completion performance for various chosen pitches of simulated few-line sensors. In practice, the pitch of the simulated sensor corresponds to the line index (out of the 64 lines in the original KITTI LiDAR sensor) of the first scan-line in the simulated few-line LiDAR sensor). We find that depth completion performance varies greatly with this choice for few-line sensors.

selection of pitch for the simulated fewer-line is critical. To closely mirror real-world settings where such few-line sensors are placed carefully, we select the best setup for each few-line sensor, yielding starting indices 53, 22, 9, and 3 for 1, 2, 4, and 8-line LiDAR, respectively. We re-emphasize that our selection of these scan-lines is not biased to any of the models we evaluate - we had used a separate architecture mimicking MIDAS, but with RGBD input, to select scan lines. Visualizations of our optimally selected scan-lines and other sub-optimally selected scan-lines are shown in Figure 6. We will release generated sparse depth maps for training and evaluation, and we hope future work compare on this same, more realistic few-line LiDAR setting.

B.3. NYUv2 Dataset

The NYUv2 dataset [32] contains 120k RGB-D images collected by a Microsoft Kinect sensor in 464 indoor scenes. We follow previous work [2, 27, 28, 35, 43] and train on 50k images from the training set and evaluate on the 654 images from the official test set. Images are downsampled and center cropped to 304x228.

C. Additional Implementation Details

We train our model with a batch size of 24 and a learning rate of 2e-4 on NYUv2, and a batch size of 8 and learning rate of 3.3e-4 on KITTI. We use the AdamW [20, 26] optimizer with weight decay 1e-2. The depth loss weight α is 0.1, 0.15, 0.25, and 0.5 for 16x, 8x, 4x, and 2x resolution decoder stages, and is decayed in the later epochs of

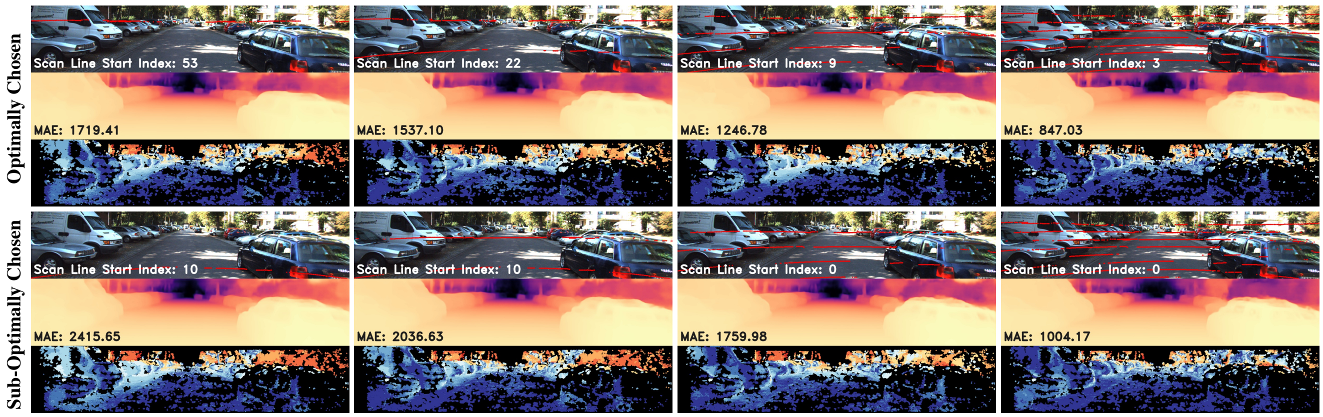

Figure 6. The chosen scan lines generally have better coverage of more diverse and distant scene elements. The depth predictions are from the MIDAS-like RGBD model used to select the scan lines. Error maps using KITTI's error color scheme are visualized below each depth map prediction.

training. The partially scale-invariant loss [9] is used for $\mathcal{D}^{initial}$. For $\mathcal{D}^{fuse}$ and $\mathcal{D}^{final}$, we use ℓ_1 for NYUv2 and both ℓ_1 and ℓ_2 for KITTI following prior work [28]. Attention using Flash Attention [6, 7] and RoPE [34] is used for both the transformer encoder and cross-attention layers [37] in the ASC module. If the input sparse depth map has more than 5500 points, which is the average # of points for 16-line LiDAR, 5500 points are independently randomly sampled for the ASC modules at each scale.

Regarding weighted sum of point features, we note that SparseFormer [38] fuses a single-channel sparse feature they interpret as "confidence." We find that because this feature does not receive direct supervision, it has little correlation with prediction quality. Thus, we add more channels and interpret it as a deep feature.

D. Extended Ablation Study

D.1. Full Ablation Study Tables

For completeness and to allow future work to fully compare with various settings of our framework, we extend our ablation studies in Tables 1, 2, 3, and 4 from the main paper with full precision and additional metrics in Tables 11, 12, 13, and 14, respectively. Note that in Table 13 we additionally include application of the NLSPN head on our pipeline with RGB input. We find that it improves performance for ResNet34+ backbone but worsens performance for ResNet34 and Effb5. As such, when using RGB-input, we do not apply add the NLSPN head.

D.2. Visualization of Weighted Sum Depth Maps

To further validate the importance of taking a weighted sum over depth errors intead of raw input depths, we visualize intermediate attention maps and and weighted sum depth

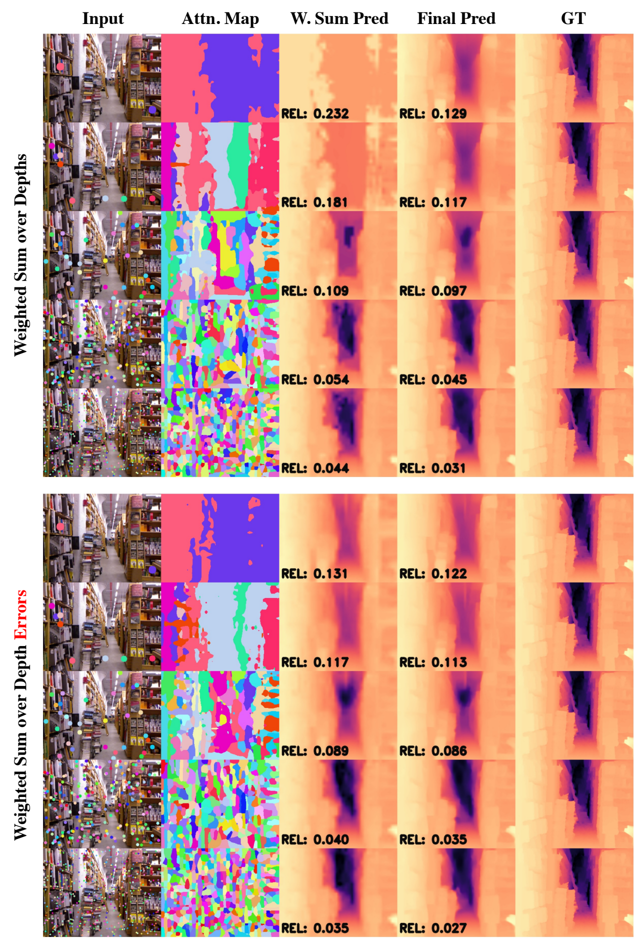

Figure 7. Visualization of intermediate attention maps and weighted sum depth predictions for models with weighted sum over features & depths and features & depth errors.

Components	# of Sampled Points														
	2			8			32			200			500		
	$\delta_{1.25}\uparrow$	REL$\downarrow$	RMSE$\downarrow$	$\delta_{1.25}\uparrow$	REL$\downarrow$	RMSE$\downarrow$	$\delta_{1.25}\uparrow$	REL$\downarrow$	RMSE$\downarrow$	$\delta_{1.25}\uparrow$	REL$\downarrow$	RMSE$\downarrow$	$\delta_{1.25}\uparrow$	REL$\downarrow$	RMSE$\downarrow$
Features	0.8849	0.1060	0.4138	0.9273	0.0764	0.3477	0.9590	0.0500	0.2678	0.9867	0.0238	0.1591	0.9930	0.0168	0.1175
Depths	0.8150	0.1361	0.5042	0.8904	0.0985	0.4056	0.9405	0.0643	0.3091	0.9681	0.0390	0.2116	0.9745	0.0331	0.1800
Depth Errors	0.8380	0.1300	0.4551	0.9146	0.0844	0.3587	0.9611	0.0496	0.2561	0.9877	0.0229	0.1499	0.9935	0.0159	0.1107
Features + Depths	0.8830	0.1075	0.4187	0.9224	0.0783	0.3539	0.9586	0.0504	0.2690	0.9856	0.0251	0.1648	0.9921	0.0183	0.1247
Features + Depth Errors	0.8709	0.1152	0.4279	0.9247	0.0782	0.3476	0.9613	0.0492	0.2612	0.9873	0.0230	0.1534	0.9934	0.0160	0.1128

Table 11. Ablation on different uses of affinity with full precision and metrics.

Components	# of Sampled Points														
	2			8			32			200			500		
	$\delta_{1.25}\uparrow$	REL$\downarrow$	RMSE$\downarrow$	$\delta_{1.25}\uparrow$	REL$\downarrow$	RMSE$\downarrow$	$\delta_{1.25}\uparrow$	REL$\downarrow$	RMSE$\downarrow$	$\delta_{1.25}\uparrow$	REL$\downarrow$	RMSE$\downarrow$	$\delta_{1.25}\uparrow$	REL$\downarrow$	RMSE$\downarrow$
Features + Depth Errors	0.8709	0.1152	0.4279	0.9247	0.0782	0.3476	0.9613	0.0492	0.2612	0.9873	0.0230	0.1534	0.9934	0.0160	0.1128
+ ω_{fuse} w/o $\mathcal{F}^{dist}$	0.8716	0.1126	0.4273	0.9234	0.0786	0.3493	0.9618	0.0485	0.2592	0.9876	0.0227	0.1523	0.9936	0.0157	0.1108
+ $\mathcal{F}^{dist}$	0.8821	0.1083	0.4131	0.9261	0.0783	0.3452	0.9620	0.0488	0.2580	0.9873	0.0228	0.1524	0.9934	0.0159	0.1118
+ Partial SI-Loss for $\mathcal{D}^{initial}$	0.8790	0.1081	0.4236	0.9287	0.0766	0.3450	0.9644	0.0472	0.2517	0.9879	0.0225	0.1492	0.9937	0.0158	0.1100
+ Initial Feature-only fusion	0.8872	0.1024	0.4094	0.9321	0.0730	0.3348	0.9632	0.0473	0.2533	0.9873	0.0231	0.1530	0.9934	0.0161	0.1121
+ ℓ_1 for $\mathcal{D}^{final}$ and $\mathcal{D}^{fuse}$	0.8946	0.0980	0.4001	0.9376	0.0688	0.3263	0.9662	0.0440	0.2450	0.9875	0.0220	0.1520	0.9934	0.0153	0.1117

Table 12. Ablation on correction confidence, initial feature-only fusion, and loss functions with full precision and metrics.

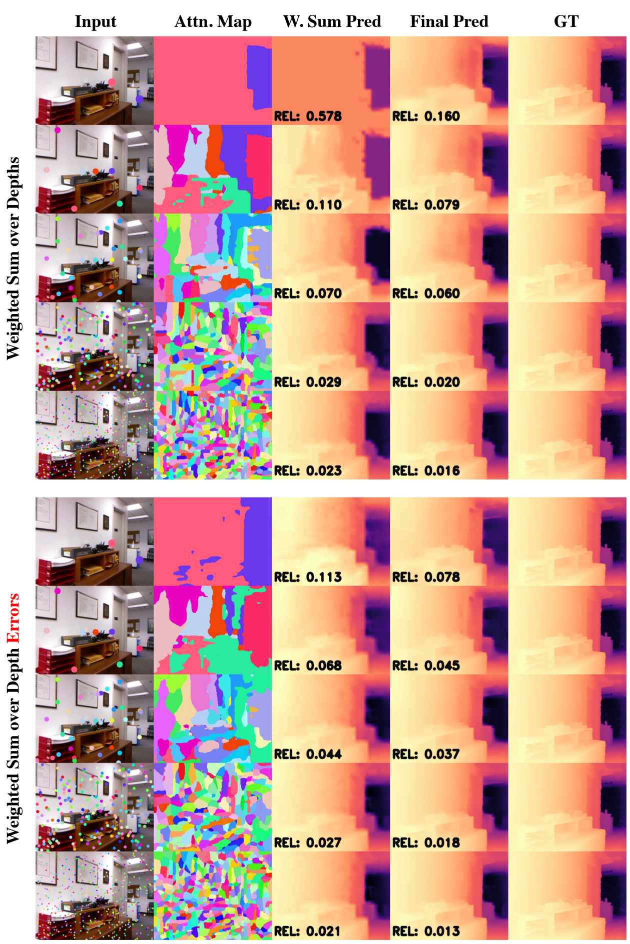

Figure 8. Additional visualizations of intermediate attention & depth maps.

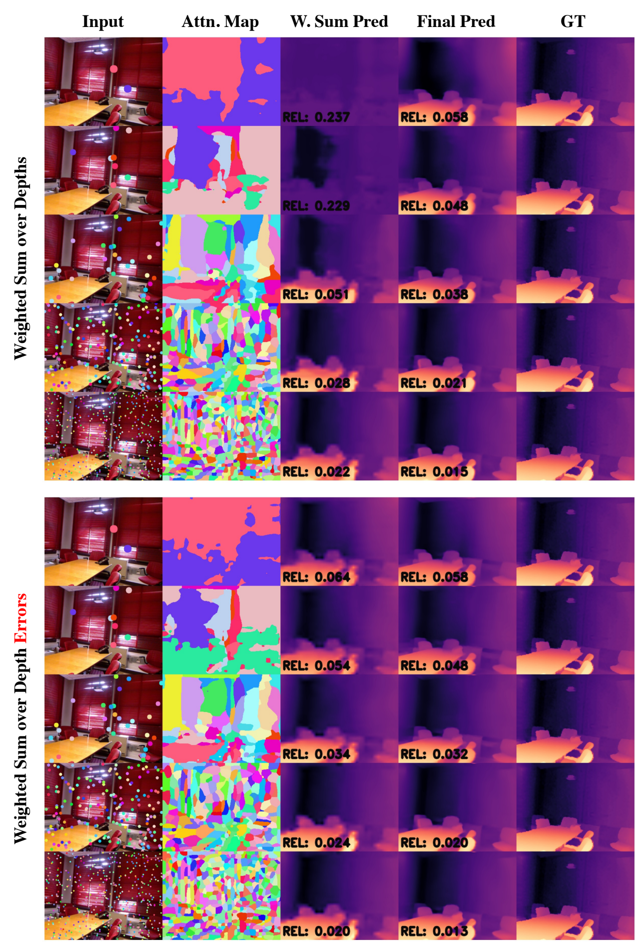

Figure 9. Additional visualizations of intermediate attention & depth maps.

Backbone	-D Input	NLSPN	# of Sampled Points														
			2			8			32			200			500		
			$\delta_{1.25}\uparrow$	REL↓	RMSE↓	$\delta_{1.25}\uparrow$	REL↓	RMSE↓	$\delta_{1.25}\uparrow$	REL↓	RMSE↓	$\delta_{1.25}\uparrow$	REL↓	RMSE↓	$\delta_{1.25}\uparrow$	REL↓	RMSE↓
Res34	✗	✗	0.8638	0.1100	0.4434	0.9192	0.0780	0.3605	0.9603	0.0470	0.2606	0.9875	0.0217	0.1512	0.9934	0.0151	0.1112
Res34	✓	✗	0.8179	0.1338	0.5078	0.9070	0.0863	0.3814	0.9617	0.0481	0.2588	0.9879	0.0223	0.1481	0.9934	0.0161	0.1113
Res34	✗	✓	0.8614	0.1112	0.4545	0.9173	0.0795	0.3709	0.9602	0.0479	0.2640	0.9869	0.0228	0.1554	0.9932	0.0162	0.1142
Res34	✓	✓	0.8154	0.1318	0.5123	0.9067	0.0842	0.3812	0.9620	0.0467	0.2572	0.9883	0.0208	0.1444	0.9939	0.0141	0.1058
Res34+	✗	✗	0.8154	0.1309	0.5034	0.9069	0.0856	0.3830	0.9612	0.0490	0.2597	0.9871	0.0232	0.1529	0.9927	0.0168	0.1169
Res34+	✓	✗	0.8210	0.1327	0.4957	0.9162	0.0805	0.3576	0.9681	0.0435	0.2369	0.9901	0.0198	0.1350	0.9948	0.0140	0.1006
Res34+	✗	✓	0.8150	0.1298	0.5055	0.9106	0.0837	0.3780	0.9640	0.0472	0.2541	0.9879	0.0225	0.1497	0.9931	0.0163	0.1142
Res34+	✓	✓	0.8350	0.1272	0.4749	0.9215	0.0783	0.3473	0.9690	0.0425	0.2335	0.9903	0.0190	0.1325	0.9951	0.0130	0.0970
Original NLSPN Model			0.8220	0.1321	0.4973	0.9117	0.0844	0.3664	0.9668	0.0444	0.2399	0.9899	0.0194	0.1349	0.9949	0.0131	0.0980
Effb5	✗	✗	0.8946	0.0980	0.4001	0.9376	0.0688	0.3263	0.9662	0.0440	0.2450	0.9875	0.0220	0.1520	0.9934	0.0153	0.1117
Effb5	✓	✗	0.8841	0.1013	0.4159	0.9336	0.0706	0.3311	0.9660	0.0445	0.2440	0.9875	0.0229	0.1512	0.9931	0.0168	0.1144
Effb5	✗	✓	0.8941	0.1000	0.4066	0.9347	0.0707	0.3299	0.9662	0.0447	0.2461	0.9874	0.0228	0.1529	0.9930	0.0165	0.1150
Effb5	✓	✓	0.8934	0.0977	0.3994	0.9363	0.0687	0.3234	0.9679	0.0425	0.2386	0.9881	0.0208	0.1461	0.9939	0.0140	0.1062

Table 13. Ablation on backbones, inputs, and NLSPN with full precision and metrics.

Components	# of Sampled Points														
	2			8			32			200			500		
	$\delta_{1.25}\uparrow$	REL↓	RMSE↓	$\delta_{1.25}\uparrow$	REL↓	RMSE↓	$\delta_{1.25}\uparrow$	REL↓	RMSE↓	$\delta_{1.25}\uparrow$	REL↓	RMSE↓	$\delta_{1.25}\uparrow$	REL↓	RMSE↓
R34+ RGBD	0.8210	0.1327	0.4957	0.9162	0.0805	0.3576	0.9681	0.0435	0.2369	0.9901	0.0198	0.1350	0.9948	0.0140	0.1006
w/ CSPN	0.8297	0.1332	0.4833	0.9204	0.0802	0.3507	0.9696	0.0428	0.2325	0.9904	0.0193	0.1328	0.9950	0.0134	0.0980
w/ NLSPN	0.8350	0.1272	0.4749	0.9215	0.0783	0.3473	0.9690	0.0425	0.2335	0.9903	0.0190	0.1325	0.9951	0.0130	0.0970

Table 14. Ablation on refinement head with full precision and metrics.

predictions for a model taking a weighted sum over features & depths and one taking a weighted sum over features & depth errors. These models correspond to rows 4 and 5 in Table 1, respectively. The visualizations are shown in Figures 7, 8, and 9. To visualize the attention map, for each pixel, we show the color of the depth point it had the highest affinity with. For weighted sum predictions, we visualize the depth map fused back into the decoder features, which is the weighted sum over input depths for the first model shown and the weighted sum over depth *errors* applied to intermediate predictions for the second model shown. Note that for fair comparison with the ablative baseline of summing over input depths, the model summing over errors does not leverage our proposed confidence correction module.

We observe that a weighted sum over depths produces depth maps of low quality, when compared to depth maps of a weighted sum over depth offsets. This can be attributed to two main factors. First, in settings with few # of points, the input points often cannot cover the entire range of depths in the scene. For example, in Figure 7 in the 2 and 8 point settings, the depths of the back of the library are not captures in the sparse input depth. As such, simply interpolating between input depth values to generate intermediate predictions is insufficient to produce reasonable depth maps as they cannot reach beyond the boundaries of the closest and furthest depths. Instead taking a weighted sum of depth errors - differences between the predicted and input depths - and applying those corrections to intermediate predictions can cover the entire range of depths in the scene.

Second, however, we find that even when the # of points increases, covering the full range of depths in the scene, the weighted sum over input depths still has errant depth values where a weighted sum over depth errors does not. More specifically, considering the 200 and 500 point settings in Figure 7, we see some artefacts near the top of the scene for weighted sum of depth predictions. This is primarily caused by errors in the cross attention. When some pixels attend to the wrong input points, potentially due to incorrect semantic groupings, taking a weighted sum over input depths directly transfers those irrelevant points' depths to those pixels, resulting in completely incorrect depth predictions. However, instead taking a weighted sum over depth errors and applying a correction to intermediate predictions, depth predictions for those pixels are still largely grounded by those pixels' intermediate depth predictions, only slightly offset potentially incorrectly by irrelevant corrections. Such small errors can more easily be corrected later on in the model when semantics become more clear, while large errors as those caused by weighted sum over input depths are more readily propagated incorrectly to final predictions. This is especially clear in settings with more # of points where the primary source of errors is from predictions at depth boundaries, where such mistakes are most common. As such, we find that taking a weighted sum of depth offsets consistently outperforms taking a weighted sum of depths es-

Method	RMSE	MAE	iRMSE	iMAE
CSPN [3]	1019.6	279.4	2.93	1.15
Sparse&Dense [18]	917.6	234.8	2.17	0.95
BDBF [30]	900.3	216.4	2.37	0.89
TWISE [17]	840.2	195.5	2.08	0.82
NConv [10]	829.9	233.2	2.60	1.03
S2D [27]	814.7	249.9	2.80	1.21
FusionNet [36]	772.8	215.0	2.19	0.93
DepthNormal [41]	777.1	235.2	2.42	1.13
DSPN [42]	766.7	220.3	2.47	1.03
MSG-CHN [22]	762.2	220.4	2.30	0.98
DeepLiDAR [29]	758.3	226.5	2.56	1.15
FuseNet [2]	752.9	221.2	2.34	1.14
ACMNet [45]	744.9	206.0	2.08	0.90
CSPN++ [4]	743.6	209.2	2.07	0.90
PointFusion [15]	741.9	201.1	1.97	0.85
NLSPN [28]	741.6	199.5	1.99	0.84
ENet [14]	741.3	216.3	2.14	0.95
GuideNet [35]	736.2	218.8	2.25	0.99
FCFRNet [24]	735.8	217.1	2.20	0.98
PENet [14]	730.0	210.5	2.17	0.94
RigNet [43]	712.6	203.2	2.08	0.90
DySPN [23]	709.1	192.7	1.88	0.82
CompFormer [44]	708.9	203.5	2.01	0.88
Ours	727.3	194.3	1.96	0.83

Table 15. Online Test Set Evaluation for 64-line KITTI.

pecially when there are more input points.

E. Extended Comparison on 64-line KITTI

In the main paper, we primarily focused on the fewer-line and variable sparsity settings for KITTI. While not our focus, we also provide online test set results for comparison. Results are in Table 15. We observe that although our ASC module is primarily developed for sparse and variable point settings, our pipeline can achieve competitive results in the 64-line online test set. Furthermore, we have shown that our module can be applied to any encoder-decoder model and is complementary to advancements in depth completion as shown through our experiments with various spatial propagation heads in the main paper.

F. Extended Comparison on 500-point NYUv2

Similarly, in addition to our extensive experiments on few and variable point settings in the main paper, we provide a comparison with existing work on the 500-point setting for NYUv2. Results are in Table 16. Our pipeline performs competitively with existing work on this largely saturated benchmark. We emphasize that in more difficult settings with fewer points and variable input distributions, our method far outperforms state-of-the-art as discussed in the main paper. Such settings are important for wider applicability of depth completion models for other datasets and tasks, and we encourage future work to evaluate on such sparser and variable distribution settings as well.

Method	$\delta_{1.25}\uparrow$	REL↓	RMSE↓
CSPN [3]	0.992	0.016	0.117
CSPN++ [4]	-	-	0.116
DeepLiDAR [29]	0.993	0.022	0.115
ACMNet [45]	0.994	0.015	0.105
Plane-Residual [21]	0.994	0.014	0.104
SparseFormer [38]	0.994	0.014	0.104
DepthCoeff [16]	0.994	0.013	0.118
DepthNormal [41]	0.995	0.018	0.112
GNN [40]	0.995	0.016	0.106
FCFRNet [24]	0.995	0.015	0.106
GuideNet [35]	0.995	0.015	0.101
PointFusion [15]	0.996	0.014	0.090
CostDCNet [19]	0.995	0.013	0.096
TWISE [17]	0.996	0.013	0.097
RigNet [43]	0.996	0.013	0.090
NLSPN [28]	0.996 (0.9955)	0.012 (0.0117)	0.092 (0.0924)
DySPN [23]	0.996	0.012	0.090
GraphCSPN [25]	0.996	0.012	0.090
CompFormer [44]	0.996	0.012	0.090
Ours	0.996 (0.9956)	0.012 (0.0115)	0.092 (0.0917)

Table 16. 500-point setting evaluation on NYUv2.

G. Comparison to Sparsity Agnostic Depth Completion

We evaluate on the sparse depth maps generated and released by SpAgNet [5]. Results are shown in Table 17. First, when trained on 500 points, our proposed framework with RGB-inpu outperforms SpAgNet is most settings. Unlike SpAgNet which does global scale correction regardless of input point location or semantic information, our pipeline considers each point individually and does semantics-guided shift corrections. Completely agnostic to the location of each point, SpAgNet is more robust to extreme distributional differences from the 500 points seen during training, and it transfers slightly better to shifted grid and 5 point settings. However, by catering to each point, our pipeline demonstrates better performance in all other settings.

Then evaluating various methods trained on 2 to 500 randomly sampled points, we find that performance improves for all settings, most notably even for the uneven shifted grid and Livox patterns, which have very different distributions compared to random sampling. This corroborates our findings from evaluating on SIFT keypoint distributions

	Method	Shifted Grid		Livox Pattern		5 Points		50 Points		100 Points		200 Points		500 Points	
		REL↓	RMSE↓	REL↓	RMSE↓	REL↓	RMSE↓	REL↓	RMSE↓	REL↓	RMSE↓	REL↓	RMSE↓	REL↓	RMSE↓
500 Points Trained	pNCNN [11]	0.519	1.922	0.061	0.333	0.722	2.412	0.108	0.568	0.061	0.338	0.040	0.237	0.026	0.170
	CSPN [3]	0.367	1.547	0.066	0.376	0.581	2.063	0.185	0.884	0.067	0.388	0.027	0.177	0.016	0.118
	NLSPN [28]	0.175	0.796	0.037	0.233	0.262	1.033	0.081	0.423	0.038	0.246	**0.019**	0.142	**0.013**	0.101
	PackNet-SAN [13]	-	-	-	-	-	-	-	-	-	-	0.027	0.155	0.019	0.120
	SpAgNet [5]	**0.110**	**0.422**	0.039	0.206	**0.131**	**0.467**	0.058	0.272	0.038	0.209	0.024	0.155	0.015	0.114
	Ours (NLSPN Base)	0.190	0.832	0.046	0.264	0.262	0.892	0.097	0.435	0.046	0.269	0.020	**0.140**	**0.013**	**0.096**
	Ours (R34 RGB)	0.131	0.539	**0.030**	**0.186**	0.145	0.584	**0.044**	**0.247**	**0.030**	**0.191**	0.022	0.149	0.015	0.110
2∼500	NLSPN [28]	0.080	0.356	0.026	0.162	0.102	0.423	0.036	0.209	**0.026**	0.168	0.020	0.134	0.014	0.101
	Ours (NLSPN Base)	0.078	0.344	**0.025**	**0.158**	0.095	**0.398**	**0.035**	**0.202**	**0.026**	**0.164**	**0.019**	**0.132**	**0.013**	**0.100**
	Ours (R34 RGB)	**0.070**	**0.332**	0.029	0.181	**0.090**	**0.398**	0.039	0.228	0.029	0.187	0.022	0.152	0.016	0.115

Table 17. Evaluation on sparse depth maps from SpAgNet [5]. Bottom three rows are trained on 2∼500 points.

Method	1 Line		4 Lines		16 Lines		64 Lines	
	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓
NLSPN	3507.7	1849.1	2293.1	831.3	1288.9	377.2	889.4	238.8
DySPN	3625.5	1924.7	2285.8	843.3	1274.8	366.4	878.5	228.6
CompletionFormer	3250.2	1582.6	2150.0	740.1	1218.6	337.4	848.7	215.9
Ours (NLSPN Base)	**3039.6**	**1365.7**	**2116.5**	**678.5**	**1206.7**	**324.2**	**818.2**	**205.3**

Table 18. Eval on various # of scan-lines on KITTI. Metric is *mm*.

that models trained on randomly sampled 2 to 500 points can transfer well even to unique patterns and distributions of input points.

H. Comparison to CompletionFormer

In this section, we additionally compare with CompletionFormer [44]. For fair comparison, we re-train on the sub-splits and sparse depth maps released by CompletionFormer. Our results are in Table 18, with baseline results taken from Table 4 of CompFormer. We find that our pipeline outperforms prior work, especially significantly for the sparsest 1-line sensor. This shows the importance of our ASC module's flexible pixel-point interaction.

I. Transfer Performance on nuScenes

In the main paper, we demonstrated quantitatively in Table 9 that 1) models with an RGBD-input encoder trained on just 64-lines on KITTI perform poorly when transferred to 32-line nuScenes [1] depth completion. 2) Our pipeline, applied to a simple ResNet34 backbone encoder-decoder with an RGB-input encoder transfers much better under similar training settings. 3) Training on variable 1 to 64 lines on KITTI yields more robust models that perform much better when transferred not only to the base 32-line nuScenes dataset but also to the simulated 8-line and 16-line nuScenes LiDAR. 4) Our pipeline with the ASC module outperforms NLSPN with or without an RGB-input encoder.

We verify these findings qualitatively in Figures 10, 11, and 12. We first notice that RGBD-input encoder models, NLSPN and Ours (NLSPN Base), trained on 64-line KITTI generate artifacts in their depth maps when applied to nuScenes as noted by prior work [39]. These artefacts follow the input distribution, indicating that 64-line trained RGBD models are not able to handle the increased distance between image pixels and depth points caused by fewer-line LiDAR and higher resolution images in nuScenes. We do note, however, that our module applied to NLSPN significantly reduces the extent of these artifacts and improves performance (MAE↓). Furthermore, we find that our ASC module applied to a standard R34 RGB-input encoder architecture transfers very well, far outperforming both RGBD models, generating largely coherent structures, and not producing any line artefacts for 32-line LiDAR. We do see performance degrade for 16 and 8 lines as the domain shift increases.

We then verify that training on variable lines on KITTI yields robust, transferable models by training all three methods on 1 to 64 lines on KITTI. We find that performance improves for all settings and that the generated depth maps are of higher quality without line artefacts. Notably, we observe that our RGB-input encoder pipeline still performs the best, demonstrating that our ASC module is able to adaptively propagate depth information even under significant domain shift. Additionally, these experiments suggest that fusing sparse depth information at the input layer, as is common in prior works, may result in worse performance when transferring to different domains, compared to using an RGB-input encoder and fusing depth later. We hope that our proposed ASC module serves as a strong baseline for further investigations in this direction.

J. Scaling Mono. Depth Estimation Models

In Tables 12 and 13 in the main paper, we show that our pipeline is complementary to larger backbones and large-scale MiDaS [31] mono-depth pretraining for both depth completion and joint estimation and completion. Notably, completion performance for sparser regimes (2 and 32 points) increases steadily as we apply the ASC module to stronger pre-trained backbones, showing that our module can effectively align these strong, context-based monocular predictions with sparse point input. On the other hand, in the dense 500 point setting where most pixels are within

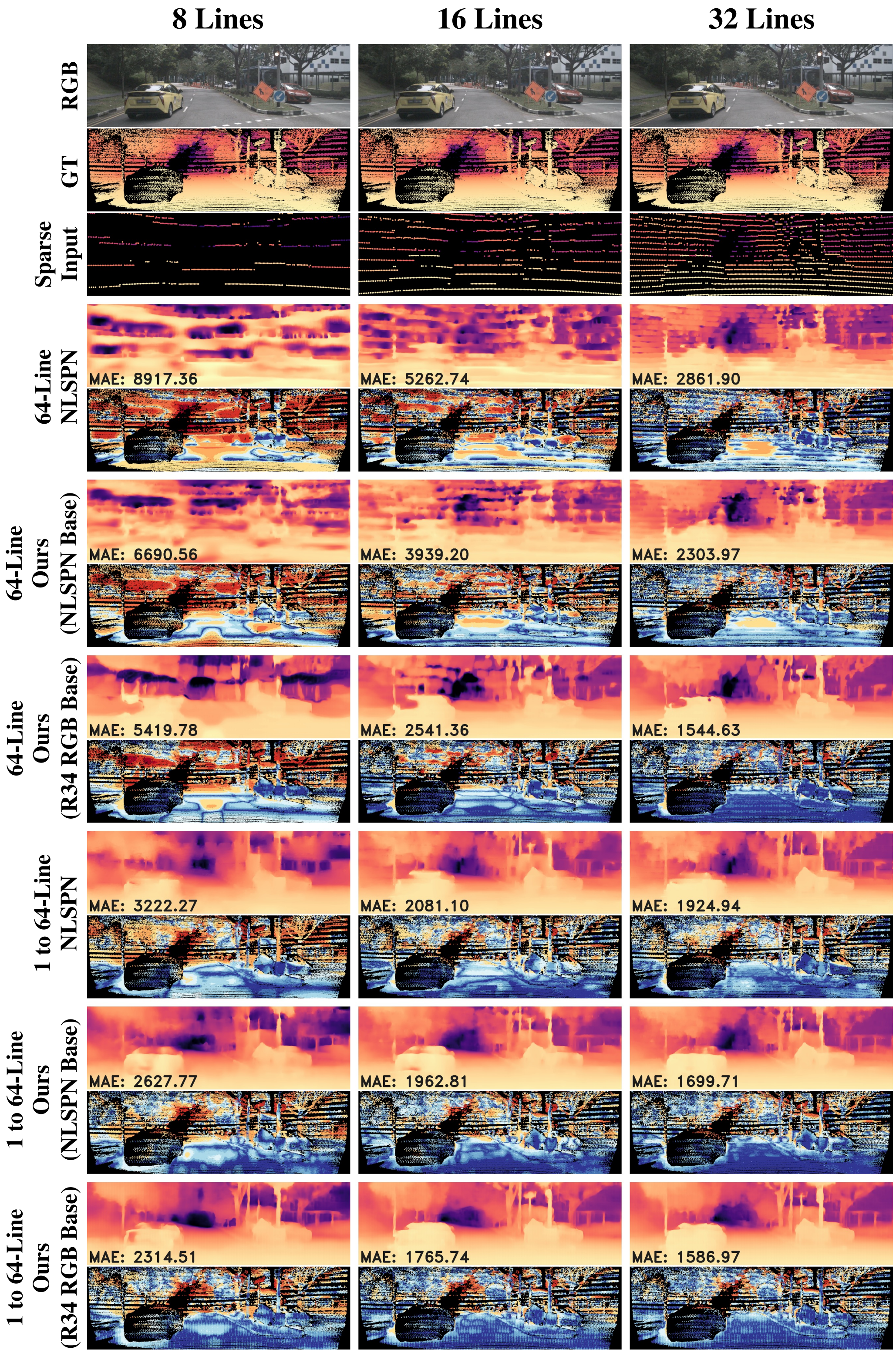

Figure 10. Domain Adaptation from KITTI to nuScenes. "64-Line" represents models trained with 64 lines on KITTI, and "1 to 64-Line" indicates the model was trained on variable sparsity, randomly sampling from 1 to 64 lines on KITTI. We emphasize that nuScenes data was not seen by any model during training. Error maps using KITTI's error color scheme are visualized below each depth map prediction.

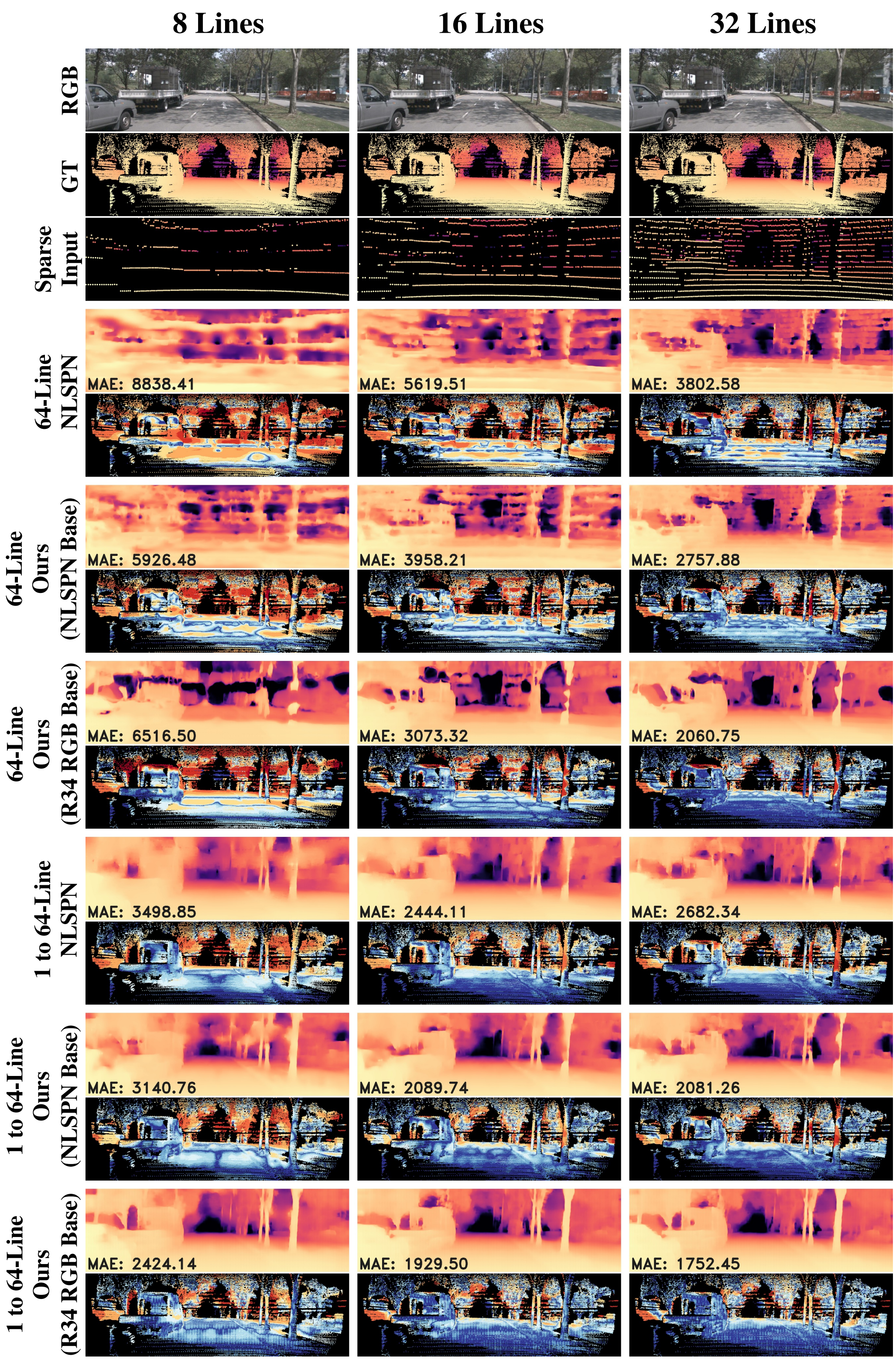

Figure 11. Additional Visualizations of Domain Adaptation from KITTI to nuScenes.

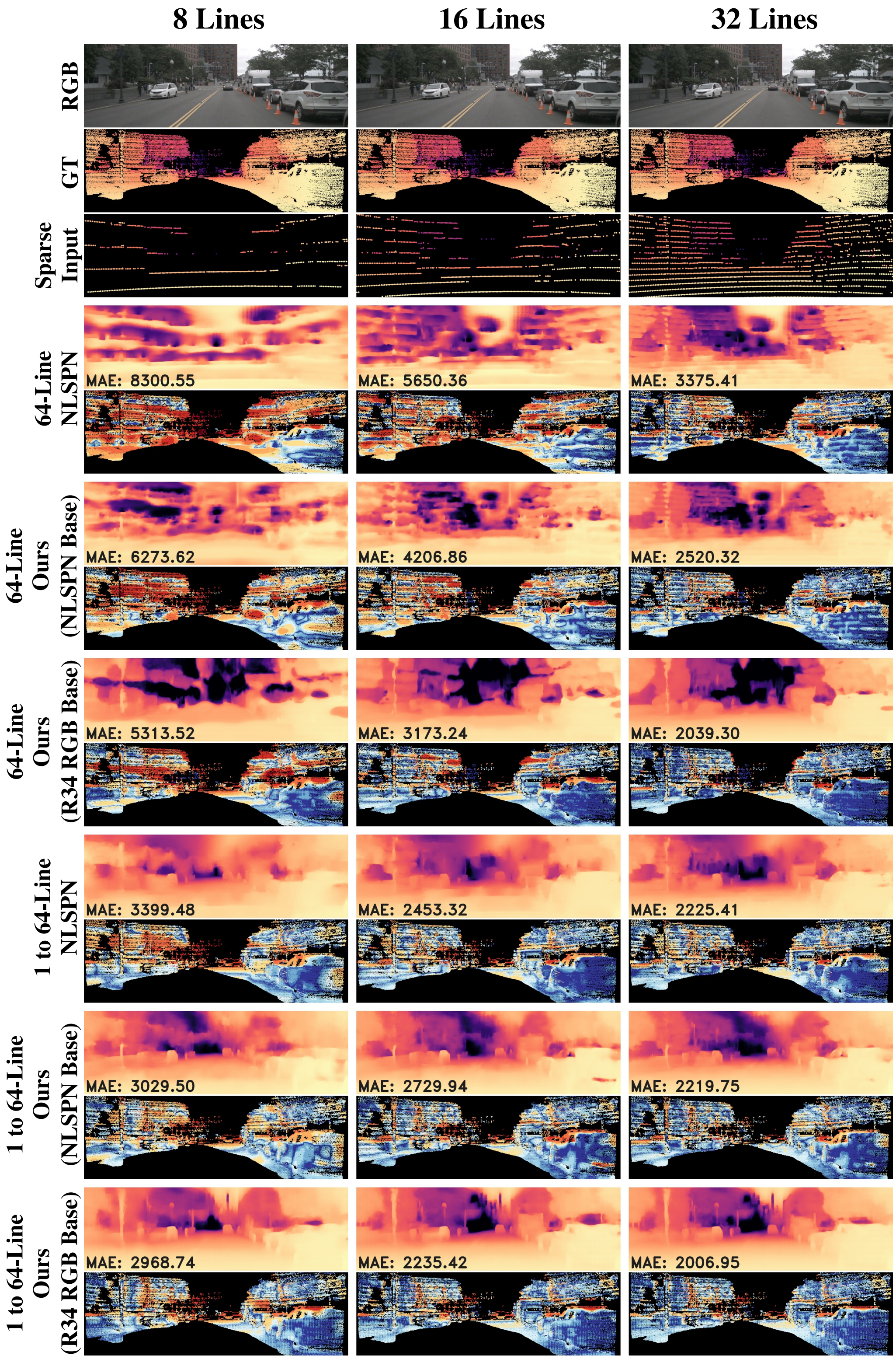

Figure 12. Additional Visualizations of Domain Adaptation from KITTI to nuScenes.

a few pixels of some input point, a simple, high-res backbone with RGBD input (ResNet34+ with RGBD) outperforms stronger backbones with RGB input (MiDaS BeiT-L). Based on our ablations in Table 3 in the main paper, this is because processing depth with the CNN encoder and maintaining higher resolution (removing initial 4x down-sampling) is crucial for this dense setting.

K. Single Pipeline for 3D Detection on Variable Scan Lines

We use our proposed depth completion model to generate dense depth maps and outproject them to a 3D point cloud. We then concatenate the original LiDAR points with the depth completed point cloud, adding a channel for a flag indicating whether the point is from the LiDAR sensor or the depth completion model. For our 3D detector, we adopt VoxelRCNN [8] for its strong performance and efficiency. Note that we took care to remove from the depth completion training set sequences geographically close to samples in the 3D detection validation set as mentioned by [33].

The results are presented in Table 19. We show mAP at moderate difficulty and 0.7 IoU threshold for the most common Car class. Echoing our analyses in the main paper, we find that our depth completion model consistently improves performance over just LiDAR at all sparsity levels. We note that in the very few-line settings of 1 or 2 lines, the LiDAR-only model largely collapses, unable to make reasonable predictions. Our completion-then-detection pipeline can detect some cars even in this setting, mainly close cars most immediately relevant for autonomous driving. Furthermore, this completion-then-detection pipeline largely maintains performance even when using a single pipeline for variable scan-lines. Finally, in the extreme case where a single pipeline is trained for 64 lines and deployed to fewer scan-lines, we find that the completion-then-detection pipeline stays far more robust than the LiDAR-only detection pipeline. We hypothesize that inputting the densified point cloud into the model leads to a far smaller domain shift in terms of point density and number between 64-line and fewer-line settings compared to just using the raw LiDAR point clouds. In all, we demonstrate that our depth completion significantly improves downstream 3D detection and can be effectively leveraged for a completion-then-detection pipeline over variable sparsities.

Training Setup	Depth Completion	1 Line	2 Lines	4 Lines	8 Lines	16 Lines	32 Lines	64 Lines
Each # of scan lines	✗	0.10	1.89	28.32	49.63	66.11	77.79	84.03
Each # of scan lines	✓	11.99	26.22	48.17	63.28	75.31	81.73	84.37
Variable # of scan lines	✗	0.03	1.49	22.91	47.74	65.46	77.63	81.19
Variable # of scan lines	✓	10.03	25.82	49.40	60.47	73.16	79.01	81.73
Only 64 lines	✗	-	-	0.51	19.89	49.90	72.24	84.03
Only 64 lines	✓	10.32	17.49	40.37	53.53	69.84	79.07	84.37

Table 19. 3D detection performance using the proposed depth completion model on KITTI.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 7

[2] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *ICCV*, pages 10023–10032, 2019. 2, 6

[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. 6, 7

[4] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10615–10622, 2020. 6

[5] Andrea Conti, Matteo Poggi, and Stefano Mattoccia. Sparsity agnostic depth completion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5871–5880, 2023. 1, 6, 7

[6] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. 2023. 3

[7] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022. 3

[8] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. *arXiv:2012.15712*, 2020. 11

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014. 3

[10] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2423–2436, 2019. 6

[11] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Michael Persson. Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12014–12023, 2020. 7

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 1

[13] Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *CVPR*, pages 11078–11088, 2021. 7

[14] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. *arXiv preprint arXiv:2103.00783*, 2021. 6

[15] Lam Huynh, Phong Nguyen, Jiří Matas, Esa Rahtu, and Janne Heikkilä. Boosting monocular depth estimation with lightweight 3d point fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12767–12776, 2021. 6

[16] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. Depth coefficients for depth completion. In *CVPR*, pages 12438–12447. IEEE, 2019. 6

[17] Saif Imran, Xiaoming Liu, and Daniel Morris. Depth completion with twin surface extrapolation at occlusion boundaries. In *CVPR*, pages 2583–2592, 2021. 6

[18] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018. 6

[19] Jaewon Kam, Jungeon Kim, Soongjin Kim, Jaesik Park, and Seungyong Lee. Costdcnet: Cost volume based depth completion for a single rgb-d image. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 257–274. Springer, 2022. 6

[20] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015. 2

[21] Byeong-Uk Lee, Kyunghyun Lee, and In So Kweon. Depth completion using plane-residual representation. In *CVPR*, pages 13916–13925, 2021. 6

[22] Ang Li, Zejian Yuan, Yonggen Ling, Wanchao Chi, Chong Zhang, et al. A multi-scale guided cascade hourglass network for depth completion. In *WACV*, pages 32–40, 2020. 6

[23] Yuankai Lin, Tao Cheng, Qi Zhong, Wending Zhou, and Hua Yang. Dynamic spatial propagation network for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1638–1646, 2022. 1, 6

[24] Lina Liu, Xibin Song, Xiaoyang Lyu, Junwei Diao, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Fcfr-net: Feature fusion based coarse-to-fine residual learning for depth completion. In *AAAI*, pages 2136–2144, 2021. 6

[25] Xin Liu, Xiaofei Shao, Bo Wang, Yali Li, and Shengjin Wang. Graphcspn: Geometry-aware depth completion via dynamic gcns. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 90–107. Springer, 2022. 6

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 2

[27] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *ICRA*, 2019. 2, 6

[28] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *ECCV*, 2020. 1, 2, 3, 6, 7

[29] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deepli-

dar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *CVPR*, pages 3313–3322, 2019. 6

[30] Chao Qu, Wenxin Liu, and Camillo J Taylor. Bayesian deep basis fitting for depth completion with uncertainty. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16147–16157, 2021. 6

[31] 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 Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1, 7

[32] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760. Springer, 2012. 2

[33] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Peter Kontschieder, and Elisa Ricci. Are we missing confidence in pseudo-lidar methods for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3225–3233, 2021. 11

[34] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 3

[35] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *IEEE Transactions on Image Processing*, 30:1116–1129, 2020. 1, 2, 6

[36] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *MVA*, pages 1–6, 2019. 6

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 3

[38] Frederik Warburg, Michael Ramamonjisoa, and Manuel López-Antequera. Sparseformer: Attention-based depth completion network. *arXiv preprint arXiv:2206.04557*, 2022. 3, 6

[39] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li Zhang. S-nerf: Neural radiance fields for street views. *arXiv preprint arXiv:2303.00749*, 2023. 7

[40] Xin Xiong, Haipeng Xiong, Ke Xian, Chen Zhao, Zhiguo Cao, and Xin Li. Sparse-to-dense depth completion revisited: Sampling strategy and graph construction. In *European Conference on Computer Vision (ECCV)*, 2020. 6

[41] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *ICCV*, pages 2811–2820, 2019. 6

[42] Zheyuan Xu, Hongche Yin, and Jian Yao. Deformable spatial propagation networks for depth completion. In *ICIP*, pages 913–917. IEEE, 2020. 6

[43] Zhiqiang Yan, Kun Wang, Xiang Li, Zhenyu Zhang, Jun Li, and Jian Yang. Rignet: Repetitive image guided network for depth completion. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 214–230. Springer, 2022. 1, 2, 6

[44] Zhang Youmin, Guo Xianda, Poggi Matteo, Zhu Zheng, Huang Guan, and Mattoccia Stefano. Completionformer: Depth completion with convolutions and vision transformers. *arXiv preprint arXiv:2304.13030*, 2023. 6, 7

[45] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing*, 2021. 1, 6