TULIP: Transformer for Upsampling of LiDAR Point Clouds (Supplementary Material)

Bin Yang[1], Patrick Pfreundschuh[1], Roland Siegwart[1], Marco Hutter[2], Peyman Moghadam[3,4†], Vaishakh Patil[2†]
[1]Autonomous Systems Lab, ETH Zurich, Switzerland, [2]Robotic Systems Lab, ETH Zurich, Switzerland
[3]School of EER, Queensland University of Technology (QUT), Australia, [4]Data61, CSIRO, Brisbane, Australia
{biyang, patripfr, rolandsi, mahutter, patilv}@ethz.ch, peyman.moghadam@csiro.au

A. Network Details

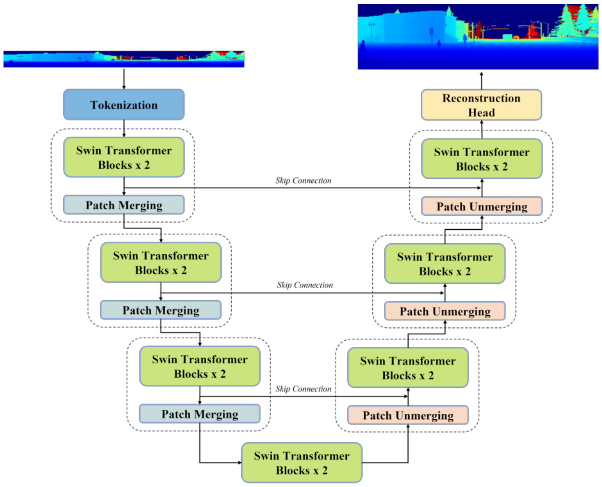

Tokenization
Reconstruction Head
Swin Transformer Blocks x 2
Skip Connection
Swin Transformer Blocks x 2
Patch Merging
Patch Unmerging
Swin Transformer Blocks x 2
Skip Connection
Swin Transformer Blocks x 2
Patch Merging
Patch Unmerging
Swin Transformer Blocks x 2
Skip Connection
Swin Transformer Blocks x 2
Patch Merging
Patch Unmerging
Swin Transformer Blocks x 2

Figure 1. **Network Architecture of *TULIP*:** The network has a symmetric design in the encoder and decoder. Each layer in the encoder consists of two Swin Transformer blocks and a patch merging layer which downscales the spatial resolution of feature maps. For the decoder, a patch unmerging layer succeeds two subsequent blocks for upscaling. A bottleneck layer is applied between the encoder and decoder to enhance the high-level feature representation. The range image is pre-processed with logarithmic transformation before being fed into the network.

A.1. Swin Transformer Block

The structure of a Swin Transformer [11] block is presented in Fig. 2. The block consists of two parts. The first part takes an input tensor with dimensions $H \times W \times C$ and initially, it performs layer normalization (LN) on the input. Then, it reshapes the feature vector into a tensor with dimensions $\frac{HW}{M^2} \times M^2 \times C$. This is achieved by partitioning the input into non-overlapping local windows of size $M \times M$, resulting in a total of $\frac{HW}{M^2}$ windows. For each of these local windows, the layer computes the query (Q), key (K), and value (V) matrices and applies the standard self-attention mechanism. The mathematical formulation is shown in Eq. 1, where B is a learnable relative positional encoding and $\sqrt{d}$ is a scaling factor. The second part has the same design and computes the attention with shifted windows.

$$Attention(Q, K, V) = SoftMax(\frac{QK^T}{\sqrt{d}} + B)V, \quad (1)$$

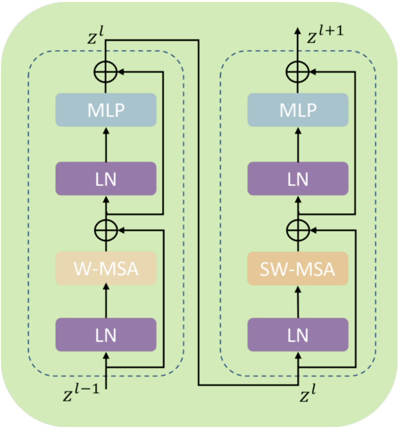


Figure 2. Details of a Swin Transformer block. MLP: Multiple Layer Perceptron, LN: Layer Normalization, W-MSA: Window Multi-Head Self Attention, SW-MSA: Shifted Window Multi-Head Self Attention

A.2. Monte Carlo Dropout

Inspired by the prior work in [12], we apply Monte Carlo Dropout (MC-Dropout) [4] to refine the prediction. For a given input x, a neural network with dropout makes a prediction $\hat{y}$ on each forward pass. Due to the active dropout layer in the inference, each forward pass effectively uses a differently configured model. By repeating the process N times ($N = 50$ in this work), it yields a set of different outputs $\{\hat{y}_1, \hat{y}_2, ..., \hat{y}_N\}$. We then compute the mean of these

[†] Authors share last authorship.

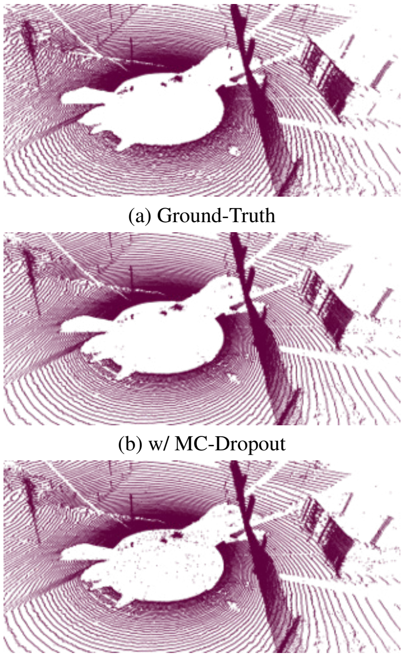
N outputs as the prediction ($\bar{y}$) and variance ($\bar{\sigma}$) which indicates the uncertainty. The formulation is shown in Eq. 2.

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} \hat{y}_i \quad \bar{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - \bar{y})^2 \tag{2}$$

In terms of LiDAR point cloud, the uncertainty can be treated as the noise in the estimation of 3D coordinates. Hence, with a pre-defined threshold parameter λ, we can remove the noisy points and obtain the final prediction ($\bar{y*}$) with the decision rule in Eq. 3. To obtain the final results, we chose a value of 0.03 for KITTI [5] and CARLA [7], and 0.0005 for DurLAR [9] dataset.

$$\bar{y*} = \begin{cases} \bar{y}, & \text{if } \bar{\sigma} < \lambda * \bar{y} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

To qualitatively show the effectiveness of post-processing the output, we visualize the range image in cases with and without MC-Dropout in Fig. 3.


(a) Ground-Truth

(b) w/ MC-Dropout

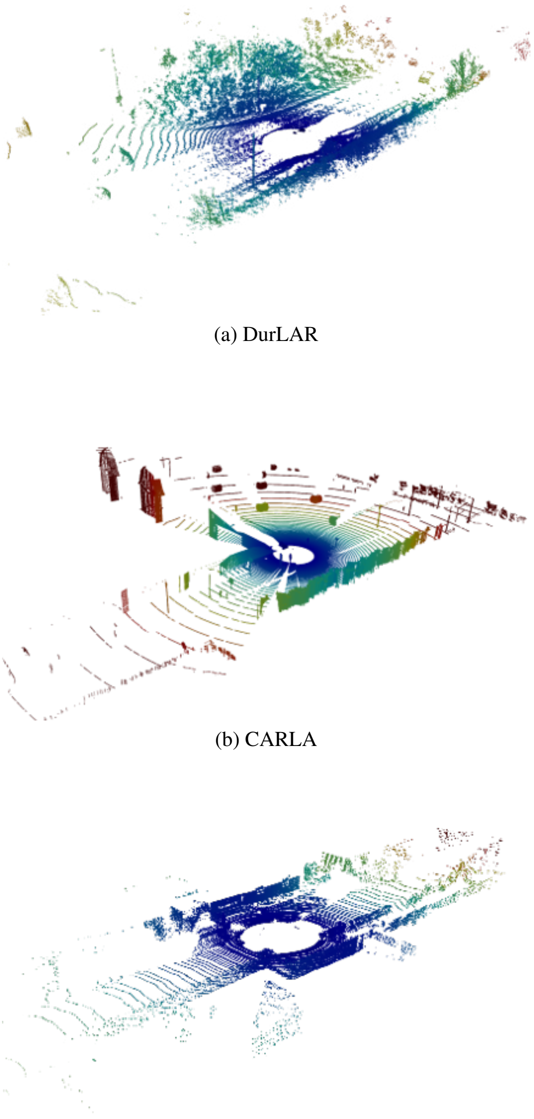
(c) w/o MC-Dropout

Figure 3. MC-Dropout cleans the range image by removing the noisy points.

B. Additional Results

B.1. Discussion

In the main experiments, our method presents a clear improvement in all three benchmarks. However, the enhancement for CARLA [7] and DurLAR [9] is not as significant as for KITTI [5] dataset. In this section, we want to elaborate on this phenomenon.


(a) DurLAR

(b) CARLA

(c) KITTI

Figure 4. Example of LiDAR point cloud (ground-truth) from the test split of each dataset. DurLAR data contains more noise in the scan pattern and CARLA data is collected in a noiseless simulation environment.

DurLAR: The dataset contains 5 sub-datasets recorded from 5 different routes. To fairly create the train and test split from the dataset, we selected the ”highway” route as the test sequence. This route contains lots of irregular objects such as trees and grass (as shown in Fig. 4), which are harder to upsample than regular objects such as houses or cars that are more frequently found in KITTI. Due to a sensor with a longer range, there are also roughly 10 times more points beyond 30 meters compared to KITTI. The point sparsity at higher ranges makes the reconstruction difficult. To support the argument, we additionally evaluated KITTI and DurLAR on points that are closer than 30 meters from the sensor origin in Tab. 1. The quantitative results indicate a similar trend of increase in Mean Absolute Error (MAE) and Chamfer Distance (CD) for the two datasets.

CARLA: The simulated point clouds in **CARLA** exhibit a high level of orderliness. This benefits the interpolation methods (e.g. ILN) that are capable of upsampling while keeping more geometrical accuracy, however, the method suffers from repeatedly upsampling points in sparse regions. Conversely, our method distributes point upsampling across the scene more uniformly. We present visualizations for CARLA in columns I-J of Fig. 6. Quantitatively, in the main expeirments of LiDAR upsampling, the competitive score in IoU but remarkably lower score in Chamfer Distance (CD) for ILN also confirms the notion that our method is superior in uniformity of upsampling, resulting in lower error in Euclidean distance.

B.2. Inference Time

We evaluate the inference time on the KITTI dataset [5]. A single forward pass for *TULIP*-L takes 1.33s. For the Monte-Carlo Dropout we run the separate forward passes in a batch and calculate the filtered result, which increases the effective inference time of *TULIP*-L to 1.61s (*TULIP* 1.13s). This is slower than ILN [7] (1.14s) and LIDAR-SR [12] (1.16s) and comparable to Swin-IR [10] (1.63s).

B.3. Model Parameters

We present the number of model parameters in Tab. 2. Swin-IR [10] has fewer parameters because it utilizes a residual network design and hence, the scale of the decoder is much smaller than our method which deploys a U-Net-based architecture. ILN [7] learns interpolation weights for neighboring points instead of upscaling the spatial resolution of features, so the network contains even fewer parameters at the cost of more memory for storing query points during training. LiDAR-SR [12] is comparable due to its similar network design as ours. It introduces a conventional CNN-based U-Net architecture for LiDAR upsampling and exceeds the total number of parameters in *TULIP*. In the main experiments, *TULIP*-L achieves more gain in reconstruction accuracy, however, the parameters are 4 times more than the baseline. A future study on shrinking network size is definitely of interest. Since we infer that most of the attention weights corresponding to those pixels with no occupancy (no beam return in 3D space) in the range image can be discarded without significantly compromising the accuracy, some model pruning techniques [6, 14] can help to reduce the training costs, in addition, they can benefit shortening inference time as well.

B.4. Generalization Capability

We test the DurLAR dataset [9] using a model trained on the CARLA [7] dataset to assess the generalization capability of different methods. The outcomes are displayed in Table 3. Both sets of results, one from training on a different dataset and the other from training on the same dataset are presented to illustrate the degeneration in upsampling performance resulting from the domain shift. Although our method still outperforms most of the other methods in this case, except that it is slightly inferior to ILN [7] in terms of IoU, we can observe significant drops in 3D evaluation metrics (IoU 17.3% and Chamfer distance 97.2%).

B.5. Different Upsampling Scales

We test the adaptability of our approach in upsampling with different scaling ratios. As it is hard to find a real-world LiDAR configuration that can scan the same scene with multiple resolutions, we use CARLA [7] dataset, created within the simulator and contains four different output resolutions: 16×1024, 64×1024, 128×2048 and 256×4096. We regard 16×1024 data as input and the other three as target resolutions for upsampling. In Tab. 4, regarding to $\times 4$ upsampling with CARLA dataset, unlike upsampling from 32 to 128 channels, Implicit LiDAR Network (ILN) [7] surpasses ours with respect to all evaluation metrics. Such an interpolation-based method tends to work more effectively for LiDAR upsampling in a noise-free environment, given that the target resolution and upsampling ratio are not exceedingly high. Raising the target resolution to 128×2048, ours is compatible with ILN, and coming to $\times 64$ upsampling, ours leads ILN by certain margins except IoU.

B.6. Downstream Tasks

Apart from the main section regarding upsampling LiDAR point cloud and evaluating the results quantitatively and qualitatively with different metrics, we want to further assess how well the upsampling process preserves the shape of foreground objects, hence we conduct an experiment using an object detection model applied to the upsampled point clouds. In particular, we assessed our method on two downstream tasks: 3D object detection and localization, which can certainly benefit from the densification of the LiDAR point cloud.

For object detection, we refer to PointPillar [8], which is one of the most commonly used models for 3D object detection. To make a fair comparison, we directly use a model pre-trained on KITTI Object Dataset [5] and generate the 3D bounding boxes on the generated, ground truth and low-resolution point clouds. For evaluation, we utilize the Average Precision (AP) with 11 precision-recall positions at an overlap threshold of 0.7 IoU and additionally calculate the mean Average Precision (mAP) over three different classes. Results are shown in Tab. 5. Compared to conventional

	KITTI [0-30m]			DurLAR [0-30m]		
Model	MAE ↓	IoU ↑	CD ↓	MAE ↓	IoU ↑	CD ↓
LIDAR-SR	0.5393	0.1041	0.1042	1.5180	0.1682	0.1079
ILN	0.9773	0.3409	0.1346	1.5672	0.3418	0.0908
TULIP (Ours)	0.4174	0.4462	0.0286	1.2408	0.3662	0.0223
TULIP-L (Ours)	**0.3678**	**0.4673**	**0.0246**	**1.1645**	**0.3758**	**0.0208**

Table 1. We evaluated upsampled point clouds within 30 meters. For Localization, we evaluated the performance of Range-MCL [1], additionally on 64x1024 ground-truth (HR) and 16x1024 input (LR) of KITTI. *The model was evaluated on the full range as requested.

Swin-IR	LiDAR-SR	ILN*	*TULIP* (Ours)	*TULIP*-L (Ours)
11.8M	34.6M	1.3M	27.1M	108.1M

Table 2. Number of parameters of state-of-art methods in LiDAR upsampling and image super-resolution. We chose the network trained on KITTI dataset.

Model	MAE ↓	IoU ↑	CD ↓
SRNO [13]	2.5704	0.1467	0.9721
HAT [2]	2.6424	0.1561	0.8667
SWIN-IR [10]	2.2080	0.1809	0.4714
LIIF [3]	1.9524	0.1757	0.1706
LIDAR-SR [12]	2.0088	0.1352	0.4076
ILN [7]	1.9044	**0.3037**	0.1291
TULIP (Ours)	**1.8468**	0.2995	**0.1256**

Table 3. Quantitative comparison of the cross-dataset experiment: results are obtained by testing DurLAR dataset with the model trained on CARLA dataset.

range image upsampling techniques [7, 12], our approach presents significantly superior results in 3D object detection. Although there remains a clear gap to the ground truth point cloud, the incorporation of an upsampling network to generate a denser point cloud proves beneficial in detecting more objects and achieving more accurate detection.

For localization, we chose RangeMCL [1], which builds an observation model formulated from the discrepancies between real and rendered range images from a mesh map for a Monte Carlo Localization framework, to recalibrate the importance of weights attributed to each particle. We followed the evaluation steps introduced in the work and assessed the localization pipeline on the point clouds upsampled from different methods. In Tab. 5, it shows that upsampling the point cloud with our method generally improves the results of localization compared to using the low-resolution one directly while ILN [7] and LiDAR-SR [12] lead to a larger error in location.

Model	MAE ↓	IOU ↑	CD ↓
Output Resolution: 64×1024			
*ILN [7]	**1.4168**	**0.3927**	**0.4447**
TULIP (Ours)	1.4776	0.3471	0.6087
Input Resolution: 128×2048			
*ILN [7]	**1.5368**	**0.3476**	0.2993
TULIP (Ours)	1.5422	0.3451	**0.2972**
Output Resolution: 256×4096			
*ILN [7]	1.6088	**0.2653**	0.2219
TULIP (Ours)	**1.5984**	0.2523	**0.1988**

Table 4. Quantitative results of scaling experiments. The input resolution is fixed with 16×1024 while the output resolution is varying. **We compare our method with Implicit LiDAR Network [7] and obtain the results using the provided pretrained model.*

C. Additional Qualitative Results

Besides more results of KITTI dataset shown in Fig 5, we provide additional visualization coming from the other two datasets in Fig 6.

References

[1] Xieyuanli Chen, Ignacio Vizzo, Thomas Läbe, Jens Behley, and Cyrill Stachniss. Range image-based lidar localization for autonomous vehicles. In *International Conference on Robotics and Automation (ICRA)*, pages 5802–5808, 2021. 4, 5

[2] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22367–22377, 2023. 4, 6, 7

[3] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628–8638, 2021. 4, 6, 7

[4] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 1

[5] 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. 2, 3, 5

[6] Woosuk Kwon, Sehoon Kim, Michael W Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. A fast post-

	Object Detection			Localization	
Model	Easy	Moderate	Hard	Location RMSE[m]$\downarrow$	Yaw RMSE[deg]$\downarrow$
Low-Resolution*	10.05	9.03	8.8	0.261	3.524
LIDAR-SR [12]	29.27	24.15	20.39	0.301	3.208
ILN [7]	38.29	28.61	23.67	0.263	3.200
TULIP (Ours)	50.23	37.57	32.12	0.238	3.015
TULIP-L (Ours)	54.15	41.33	37.19	0.238	2.981
High-Resolution*	73.33	62.78	59.63	0.232	2.262

Table 5. We evaluated a pretrained Pointpillars model [8] and RangeMCL [1] model on ×4 upsampled point clouds from the KITTI validation dataset [5] and report the overall results (averaged over classes 'Car', 'Cyclist' and 'Pedestrian' for object detection). Point clouds are generated by each model pretrained on KITTI Raw Dataset respectively. *Input and ground-truth data are tested as well.

training pruning framework for transformers. *Advances in Neural Information Processing Systems*, 35:24101–24116, 2022. 3

[7] Youngsun Kwon, Minhyuk Sung, and Sung-Eui Yoon. Implicit lidar network: Lidar super-resolution via interpolation weight prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8424–8430. IEEE, 2022. 2, 3, 4, 5, 6, 7

[8] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 3, 5

[9] L. Li, K.N. Ismail, H.P.H. Shum, and T.P. Breckon. Durlar: A high-fidelity 128-channel lidar dataset with panoramic ambient and reflectivity imagery for multi-modal autonomous driving applications. In *Proc. Int. Conf. on 3D Vision*. IEEE, 2021. 2, 3, 7

[10] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 3, 4, 6, 7

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1

[12] Tixiao Shan, Jinkun Wang, Fanfei Chen, Paul Szenher, and Brendan Englot. Simulation-based lidar super-resolution for ground vehicles. *Robotics and Autonomous Systems*, 134: 103647, 2020. 1, 3, 4, 5, 6, 7

[13] Min Wei and Xuesong Zhang. Super-resolution neural operator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18247–18256, 2023. 4, 6, 7

[14] Paul Wimmer, Jens Mehnert, and Alexandru Condurache. Interspace pruning: Using adaptive filter representations to improve training of sparse cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12527–12537, 2022. 3

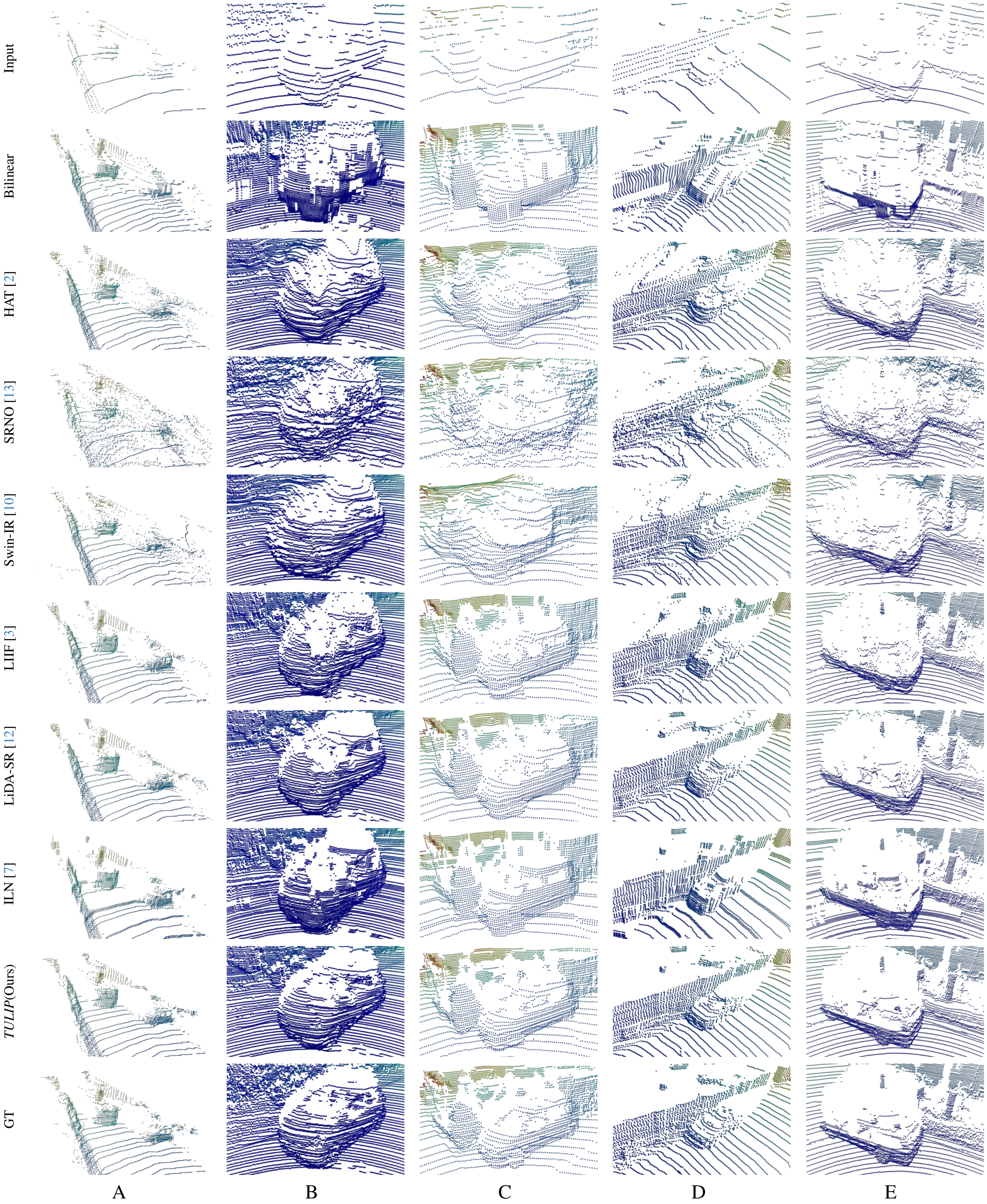


Figure 5. Additional Qualitative Results of KITTI Dataset with full comparison: Our approach outperforms all state-of-the-art approaches in upsampling the point cloud in a geometry-aware manner, specifically in terms of reconstruction of objects like cars, walls, and lidar sweeps while producing much fewer noisy points.

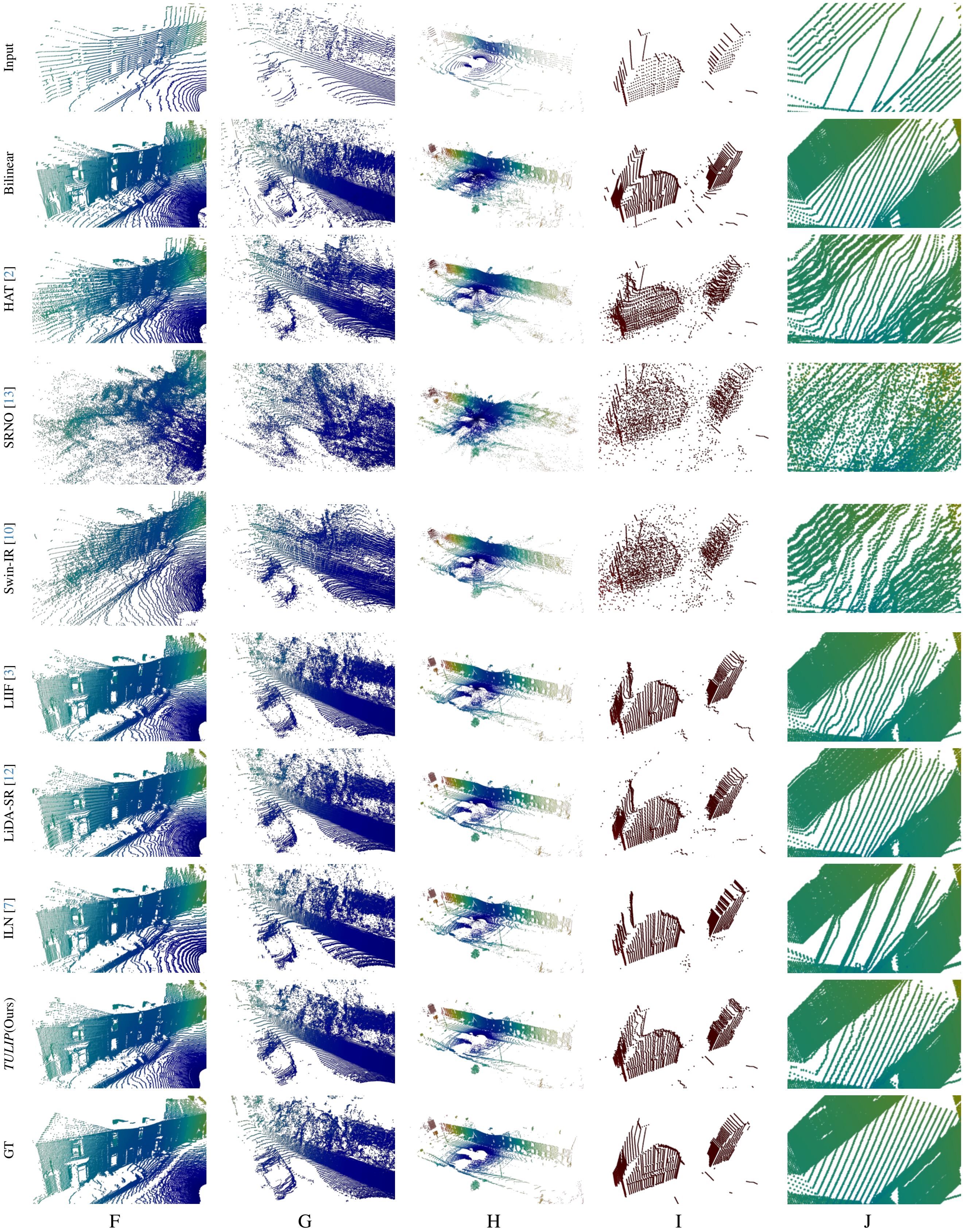


Figure 6. Qualitative Results on DurLAR [9] (**F-H**) and CARLA [7](**I-J**). Our approach can outperform state-of-the-art methods in upsampling scene-related contexts with complex and simple geometry and under both noisy and noiseless circumstances.