Appendix of WeatherGen: A Unified Diverse Weather Generator for LiDAR Point Clouds via Spider Mamba Diffusion

A Appendix

In this Appendix, we provide additional information. In Section A.1, we introduce the process of projecting LiDAR point clouds onto range maps; In Section A.2, we introduce more detailed architectures of spider mamba and latent feature aligner; In Section A.3 we introduced the setting of training steps; In Section A.4, we provide analysis and visualization comparison between generations of different denoising steps; In Section A.5, we provide visual comparisons with other methods in promoting 3D object detection; In Section A.6, we show the capability of WeatherGen in handling the LiDAR point cloud densification task; In Section A.7, we introduce the different parameter settings in map-based data producer, as well as the customized generation results; In Section A.8, we present more generation results under diverse weather conditions, including clear weather, snowy, foggy, and rainy days. In Section A.9, we discuss the limitations and broader impacts of WeatherGen and provide possible solutions.

A.1 LiDAR Data Representation

In outdoor scenes, LiDAR point clouds are often very sparse and irregular. The range map has been proven to be a successful representation of LiDAR data, overcoming the sparsity and irregularity issues of LiDAR point clouds [2, 8, 9]. In the task of LiDAR point cloud generation, recent methods also utilize range maps as the representation form of LiDAR data [10–13, 15, 17].

The range map is a representation method with virtually no information loss [9]. As shown in Figure 1, we transform the coordinate space of each point, projecting it from Cartesian coordinates $\mathbf{r} = (x, y, z) \in \mathbb{R}^3$ to spherical coordinates $\mathbf{s} \in (\theta, \phi, d)$:

$$\begin{aligned} d &= \sqrt{x^2 + y^2 + z^2}, \\ \theta &= \texttt{arccos}(z/\sqrt{x^2 + y^2 + z^2}), \\ \phi &= \texttt{atan2}(y, x), \end{aligned} \tag{1}$$

where θ is the inclination, ϕ is the azimuth, d is the depth. d is further converted into the logarithmic representation $d = \texttt{log}(d+1)/\texttt{log}(d_{max}+1) \in [0,1]^{H\times W}$. We can obtain a two-channel range map by performing the same operation on the intensity channel and the depth channel. For a 64-beam LiDAR, we set the dimensions of both depth and intensity to 64×1024, making this an almost lossless data transformation method. A row in the range map corresponds to a beam circle obtained from a full rotation scan by the LiDAR sensor, and a column in the range map corresponds to the central ray acquired from the emission of the LiDAR sensor.

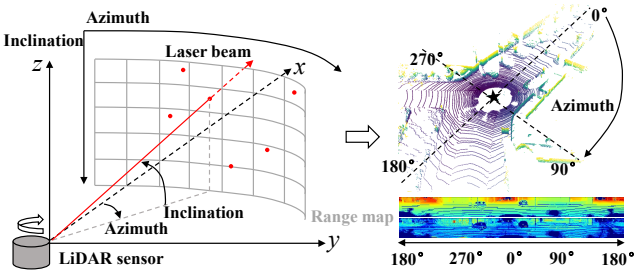

Figure 1. The conversion process from the LiDAR point cloud to the range map.

A.2 More Detailed Architectures

In Figure 2, we show additional visualization details of the spider mamba and latent feature aligner to gain a more intuitive understanding of the structure. The content displayed in Figure 2 matches the descriptions in the corresponding part of the main text. The spider mamba models feature interactions along the LiDAR beam circles and central rays, which correspond exactly to the rows or columns of the range map and are more consistent with the imaging method of LiDAR. The latent feature aligner can pull the generated results closer to real-world data in the latent space, which can greatly expand the domain boundaries of small-scale real-world data, and is beneficial for learning a broader and more universal set of real-world knowledge [14, 16].

The process of fusing the time step embedding $\mathbf{t}$ and the weather control signal $\mathbf{w}$ is straightforward like the Group Normal. We first project $\mathbf{t}$ and $\mathbf{w}$ into the same space through the linear layer, and then, we chunk them into the

scale and shift and integrate them into the intermediate features $\mathcal{H}$, the process is following:

$$\mathcal{H}_{output} = (\mathcal{H}_{input} \cdot (\mathbf{t}_{scale} + \mathbf{w}_{scale})) + (\mathbf{t}_{shift} + \mathbf{w}_{shift}). \tag{2}$$

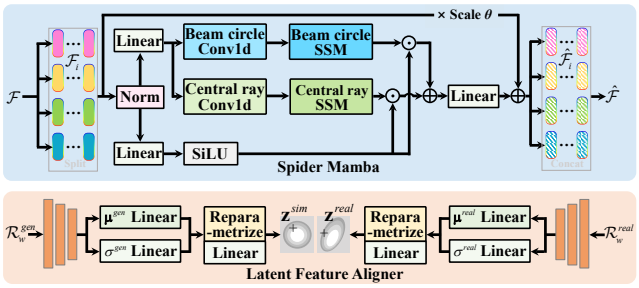

Figure 2. More detailed architectures of the spider mamba and latent feature aligner.

Table 1. Quantitative comparisons with different fine-tuning steps on Seeing Through Fog [1] under snow heavy test split.

Fine-tuning steps	Point cloud	Range map	BEV occupancy grid	
	FPD↓	FRD$\times 10^{1}$ ↓	MMD$\times 10^{-4}$ ↓	JSD$\times 10^{-1}$ ↓
50,000	68.19	143.09	1.94	0.83
100,000	61.28	123.59	1.72	0.89
150,000	59.28	124.17	1.71	0.77
200,000	58.74	124.90	1.64	0.72
250,000	58.32	129.14	1.84	0.75
300,000	58.02	124.97	1.80	0.82

A.3 Training Steps Analysis

Our model is first trained on KITTI-360 for 300,000 steps with a learning rate of 0.0004 and a batch size of 8. If the device allows, a larger batch size is beneficial. After the pre-training, we no longer use map-based data producer but directly fine-tune with the small-scale diverse weather LiDAR data provided by Seeing through fog [1] for 150,000 steps, with a learning rate of 0.0001 and a batch size of 8. The whole training time on 4 RTX 3090 GPUs is around 100 hours. It needs to be emphasized that the diverse weather LiDAR data provided by Seeing through fog is on a small scale and insufficient to support direct training, the relevant experimental results can be found in the "Ablation study " of the main text. Table 1 presents the experimental results for different fine-tuning steps. It can be observed that after 150,000 steps, the improvement in metrics is very limited. Setting the steps of fine-tuning to 150,000 strikes an optimal balance between performance and training time.

A.4 Denoising Steps Analysis

In Figure 3, we carry out an unconditional generation denoising steps analysis for four methods and present the visualization effects of WeatherGen at different denoising steps. It can be observed that from 128 steps to 256 steps, the model improvement is relatively small. The performance enhancement becomes quite limited when increasing from 256 steps to 1024 steps. Despite achieving minor improvements, the time consumption has quadrupled. Utilizing the results from 256 steps is more sufficient. Using 256 steps is the most appropriate approach.

A.5 Visual Comparisons with Other Methods in Promoting 3D Object Detection

In this section, we provide a visual comparison with other competitive methods LSS [4], FSRL [3], and LISA [5] in terms of promoting 3D object detection in snow, fog, and rain weather scenes. From Figure 4, we can see that the mini-weather dataset created using our WeatherGen can more effectively promote the downstream task, resulting in a significant improvement in detection accuracy, while other methods exhibit more missed and false detections.

A.6 LiDAR Point Cloud Densification

Densifying existing sparse LiDAR point clouds (less than 64 beams) also can significantly reduce data collection costs. WeatherGen is also capable of this task, we conduct a comparison with the most competitive R2DM [11] on top of Repaint [7]. For effectively evaluating, we employ RangeNet [9] for semantic segmentation on both the initial and the densification, aiming to observe whether the consistency in semantic information is well preserved. In addition to observing the completeness and realism of the LiDAR point clouds, we also need to verify the rationality. As shown in Figure 5, we present the original LiDAR point clouds and their segmentation results. The first row shows the original LiDAR point cloud, the semantic segmentation results, and the point cloud absence at three different levels of severity. Immediately below, we show the completion results from various methods along with their corresponding semantic segmentation results. Among them, 10 % reserved (where 90% of the points are randomly removed) is the most challenging scenario, while 25% reserved (16-beam) and 50% reserved (32-beam) are more common scenarios. R2DM [11] and Text2LiDAR [15] encounter extensive errors when completing roads, while WeatherGen, benefiting from the spider mamba mechanism, modeling the feature interactions in a per-beam manner, effectively densifies the road surface information. It is evident that the densification by WeatherGen aligns more closely with the characteristics

of the real, and the generation of semantic information is more complete and accurate.

Figure 6 presents more densification results. It can be observed that WeatherGen's ability to complete details is quite outstanding, which provides opportunities for densifying existing low-density data.

A.7 Customized Diverse Weather LiDAR Point Clouds Generation

WeatherGen supports LiDAR data generation under any severity level of weather conditions. Since the data in Seeing Through Fog is not collected under extremely severe weather conditions, the data in our experiments are all obtained with the map-based data producer parameters set to common. Snow, fog, and rain have their respective r_w, `BDF`'s droprate, and the number of noise points in $\mathcal{R}_n$ set to $\{0.13, 0.60, 1000\}$, $\{0.24, 0.50, 1800\}$, $\{0.20, 0.30, 100\}$. No matter which parameter setting is used, the generation results for each type of weather data are diverse. Figure 7 shows the generation effects under different degrees of adverse weather conditions. When generating data for extremely adverse weather conditions, due to the lack of real-world collected data, we will not employ subsequent fine-tuning. In theory, we can generate an infinite variety of weather conditions for LiDAR data, which can be customized for different application regions.

A.8 More Visualizations of Diverse Weather Generations

We provide more visualizations of LiDAR point cloud data generated under diverse weather conditions, such as clear, snow, fog, and rain. The generation under clear weather conditions is the unconditional generation. Figure 8, Figure 9, Figure 10 and Figure 11 showcase more generation results and details. As we can see, WeatherGen accurately reflects the characteristics of LiDAR point clouds under diverse weather conditions, including varying degrees of noise and missing data at different distances. Under clear weather conditions, the generated results are very diverse, with generated objects having clear edges and contours. Under snowy conditions, the generated LiDAR data exhibits properties characteristic of snow, with some noise points appearing. Since snow often accompanies wet ground, there is a situation of partial point drop on the ground, which is consistent with the characteristics of real-world data. Under foggy conditions, the generated LiDAR point cloud shows more significant attenuation at a slightly farther distance, with more point drops occurring. Targets in the scene also exhibit unclear contours due to the attenuation effect brought by fog. Under rainy conditions, due to the irregular refraction of raindrops, a small number of noise points appear, and more characteristically, the wet ground brings about point drops in the generated results. These are all consistent with the characteristics of real-world diverse weather LiDAR data, further proving the superiority of WeatherGen.

A.9 Limitations and Broader Impacts

Despite WeatherGen achieving excellent results in the diverse weather generation, it still has some shortcomings. The main issue is that WeatherGen lacks the capability to generate scenes containing rare targets effectively. Secondly, WeatherGen is not yet able to accurately generate LiDAR data for rare weather conditions, such as sleet. Collecting more data that includes rare objects and data from rare weather conditions is very beneficial for the model in overcoming the difficulties mentioned above.

In terms of broader impacts, WeatherGen is very likely to be customized for large cities to promote the rapid development of autonomous unmanned devices, which will further increase the imbalance in urban development, hinder the sharing of technology, and cause potential social injustice. To guard against such situations, a relevant usage scenario survey can be conducted, and appropriate usage permission restrictions can be implemented. Furthermore, when creating a diverse-weather LiDAR dataset using WeatherGen, training or guidance from professionals is required. Otherwise, it may lead to low-quality data annotation and sample imbalance, potentially contaminating existing datasets [6].

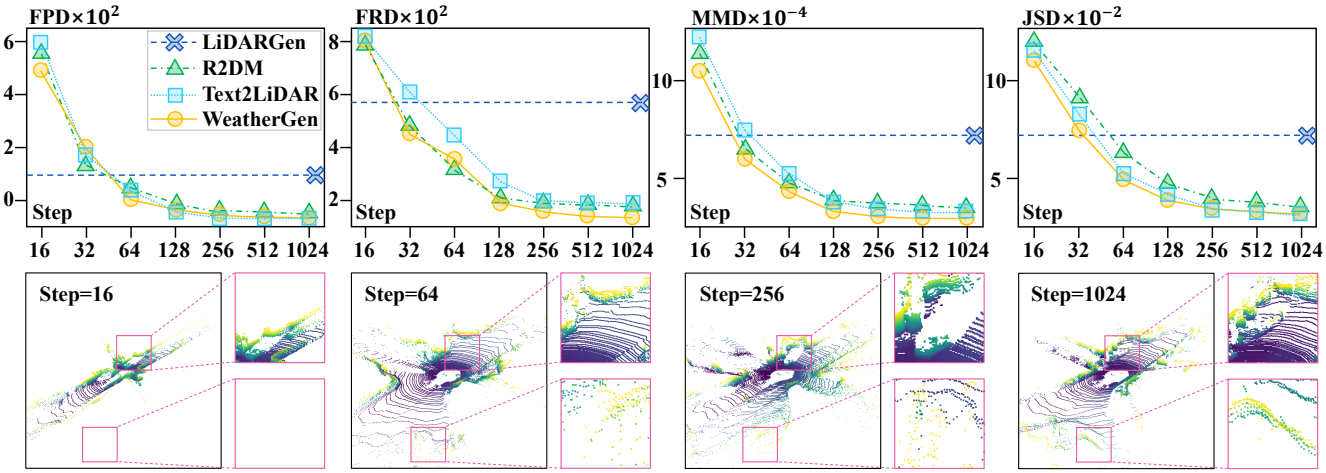

Figure 3. Denoising steps analysis and visualization comparisons.

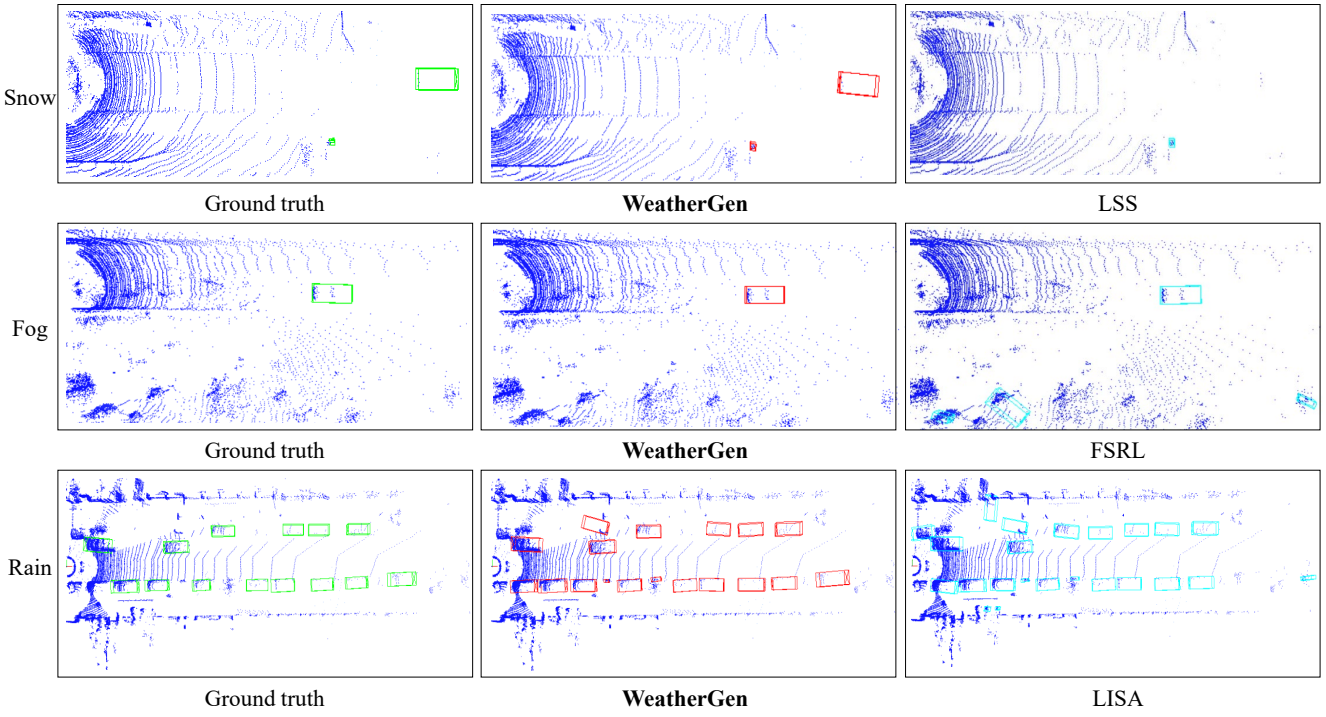

Figure 4. Visual comparisons with other competitive methods in promoting 3D object detection.

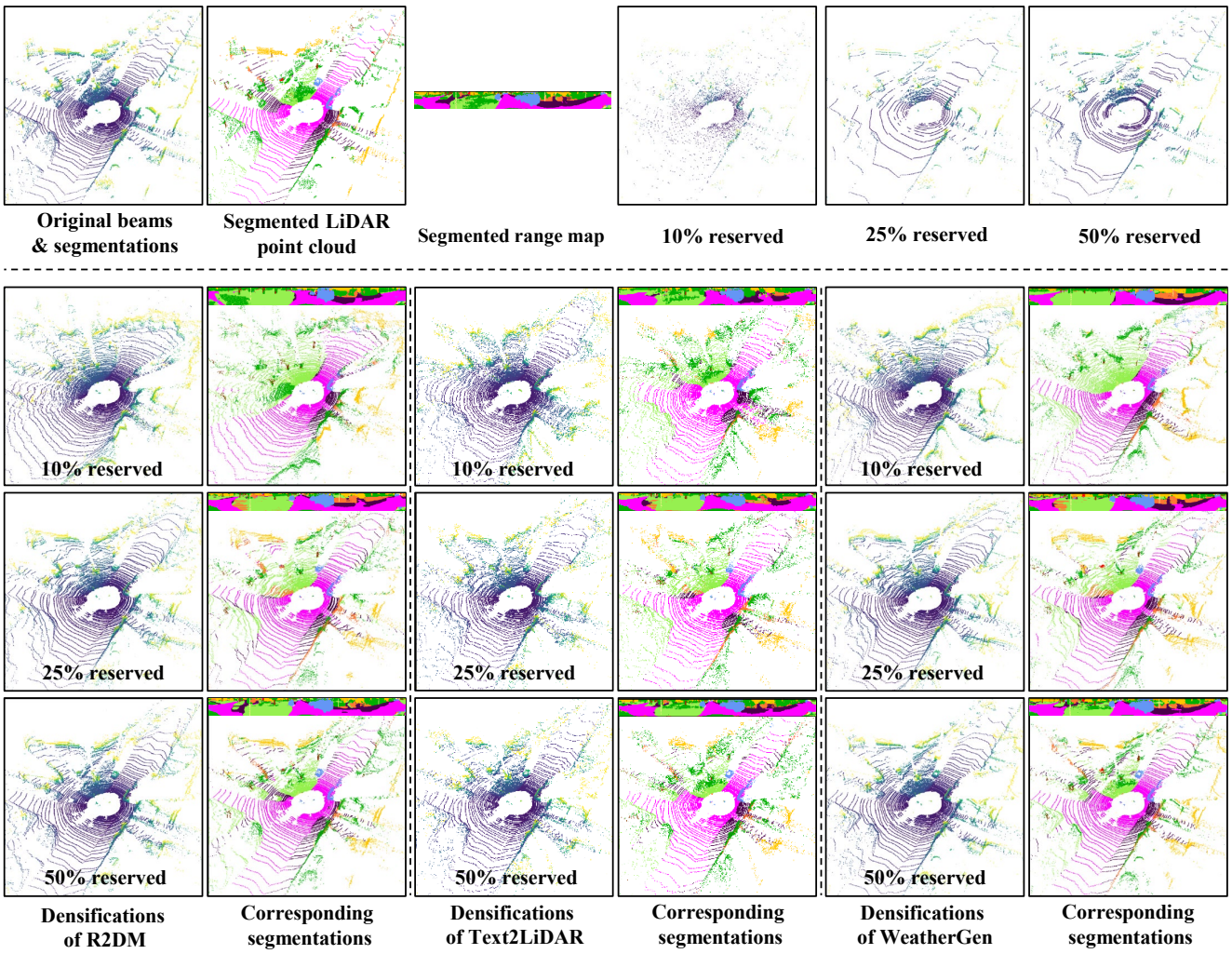

Figure 5. Visual comparisons of LiDAR point cloud densification task with R2DM.

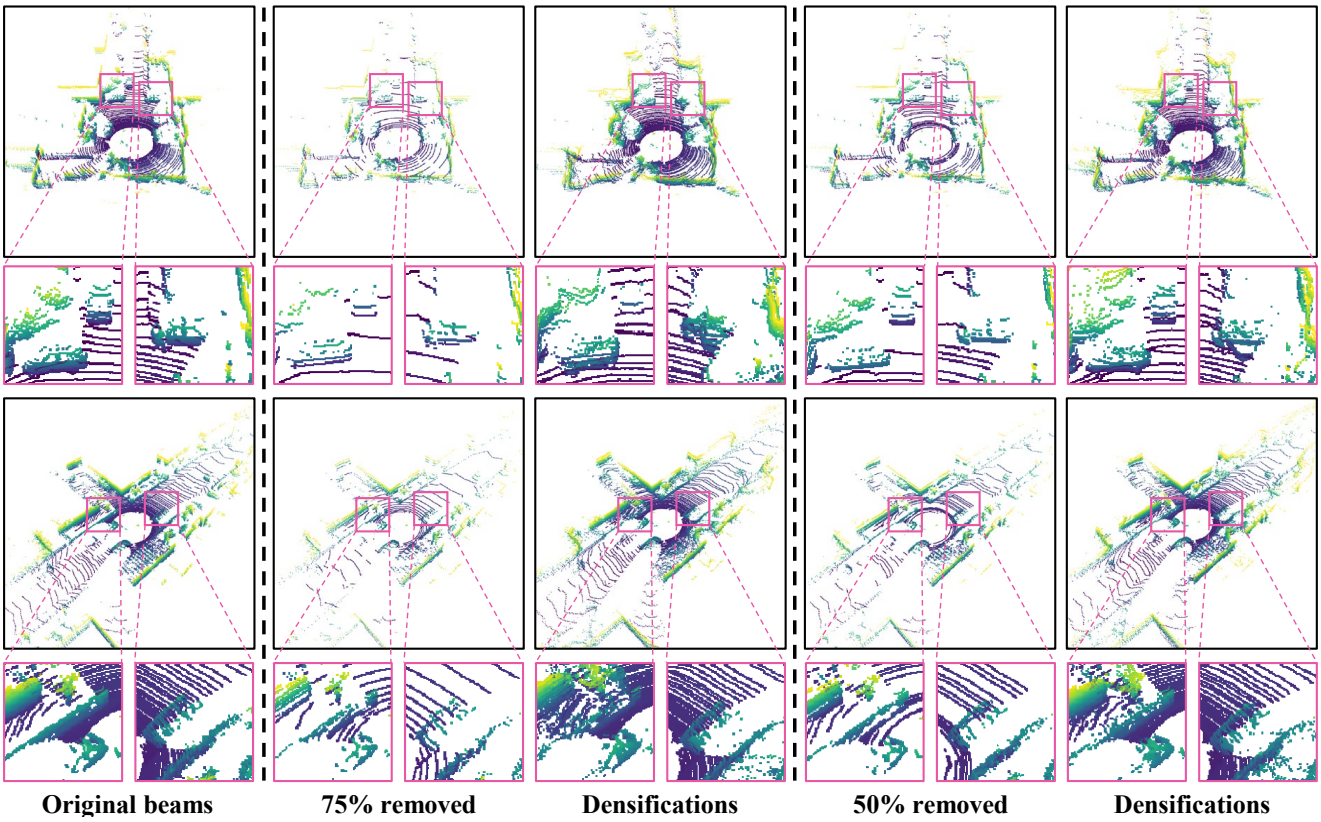

Figure 6. More LiDAR point cloud densification results of WeatherGen.

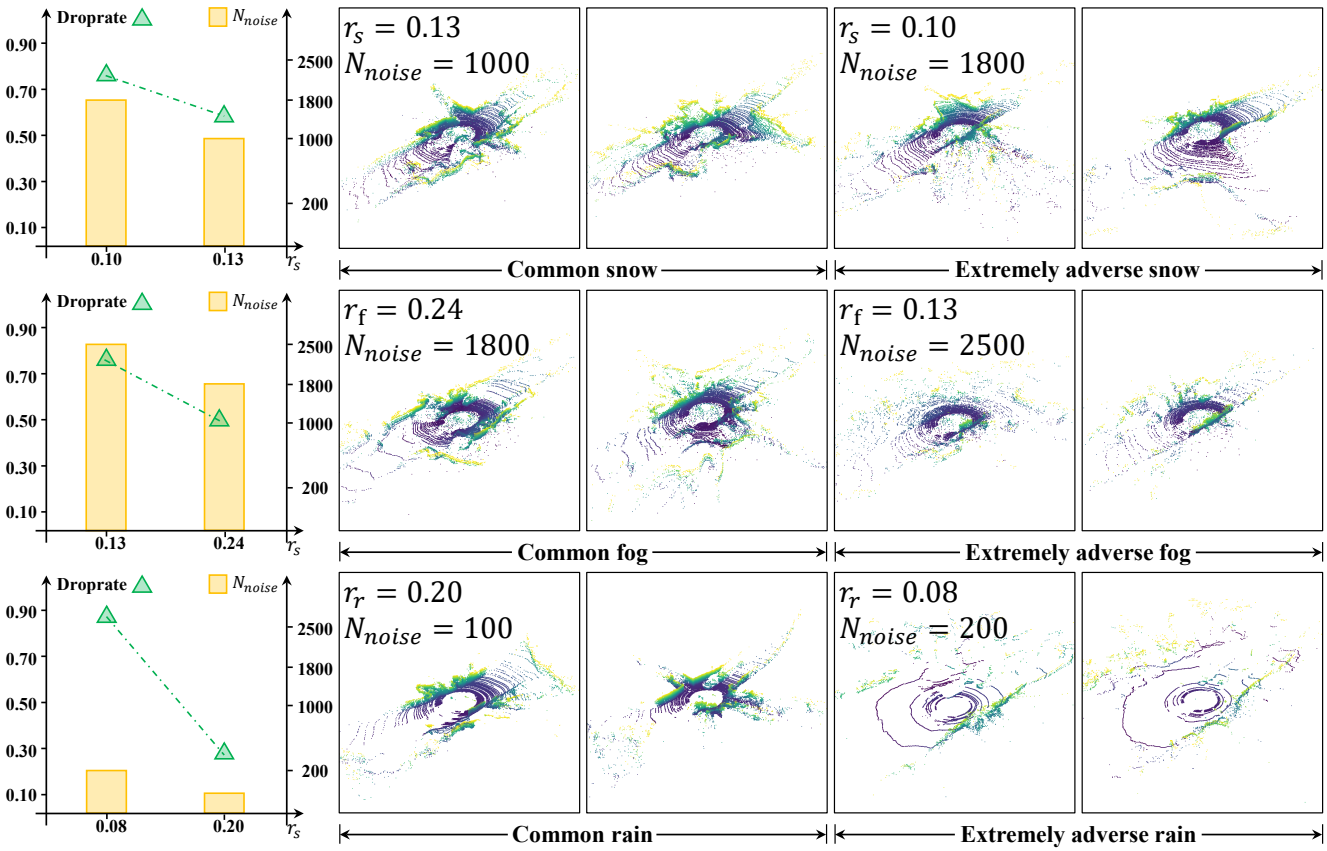

Figure 7. Different-degree weather-conditioned generations and parameter trends.

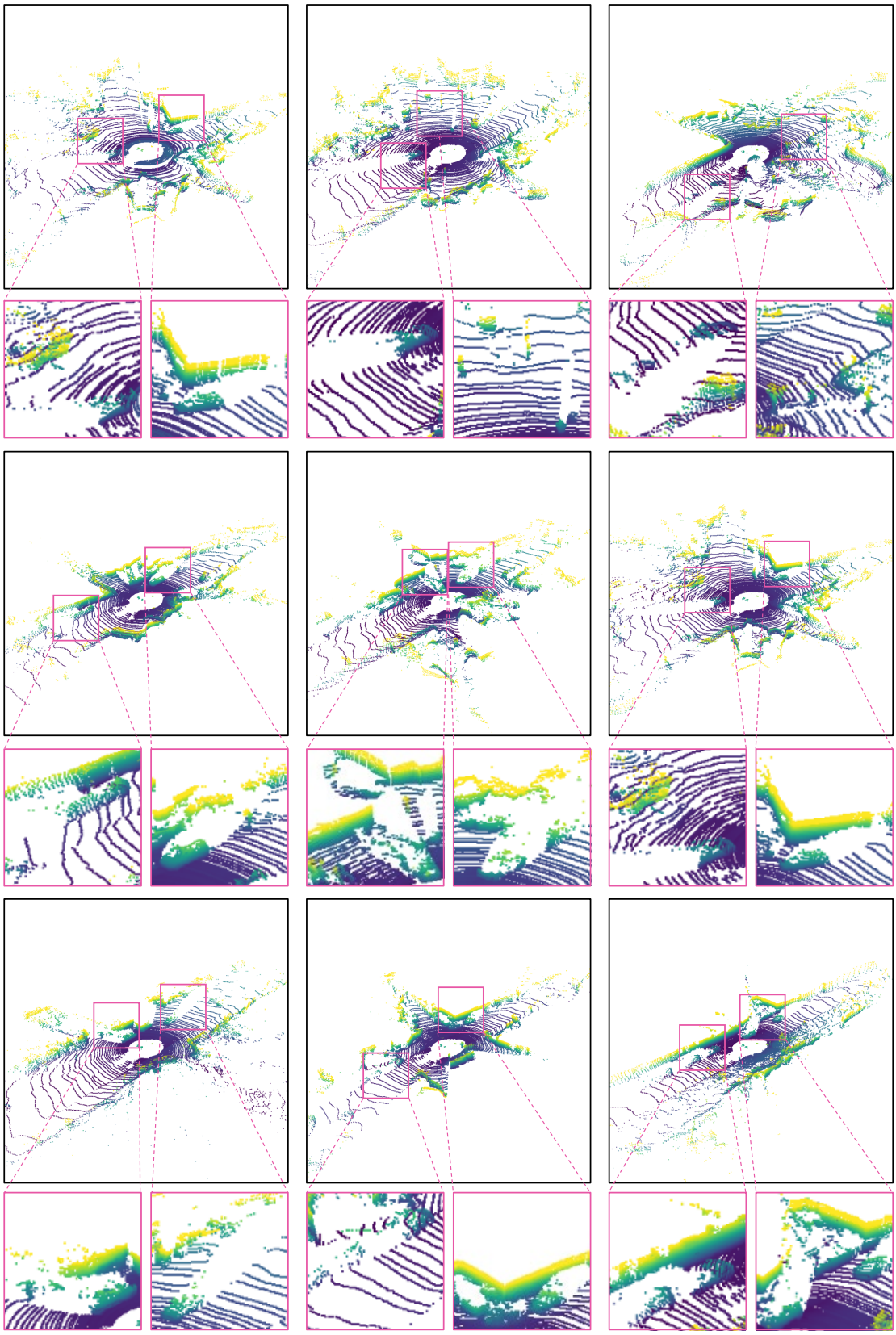

Figure 8. Visualizations of LiDAR point cloud generated under clear weather conditions.

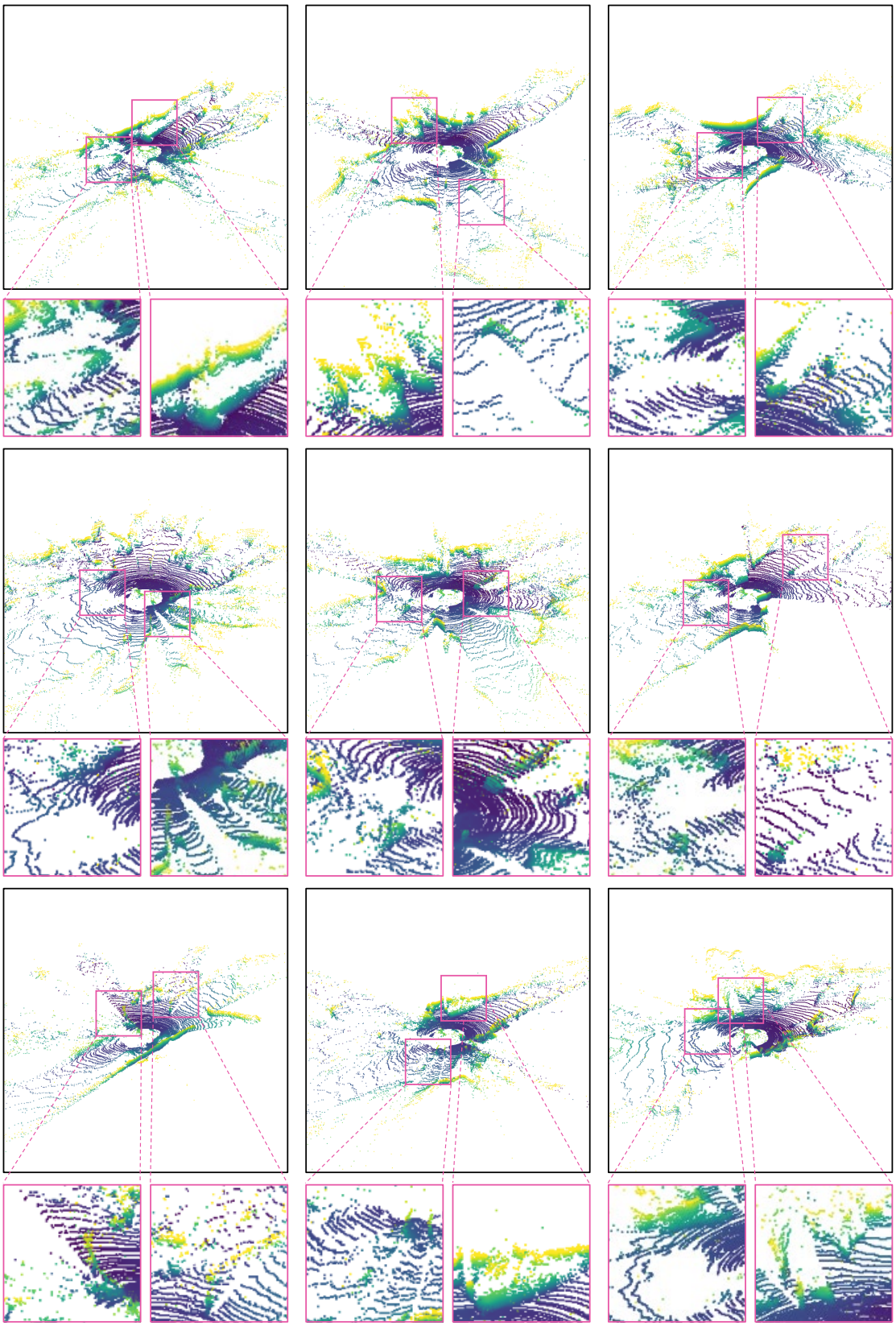

Figure 9. Visualizations of LiDAR point cloud generated under snowy weather conditions.

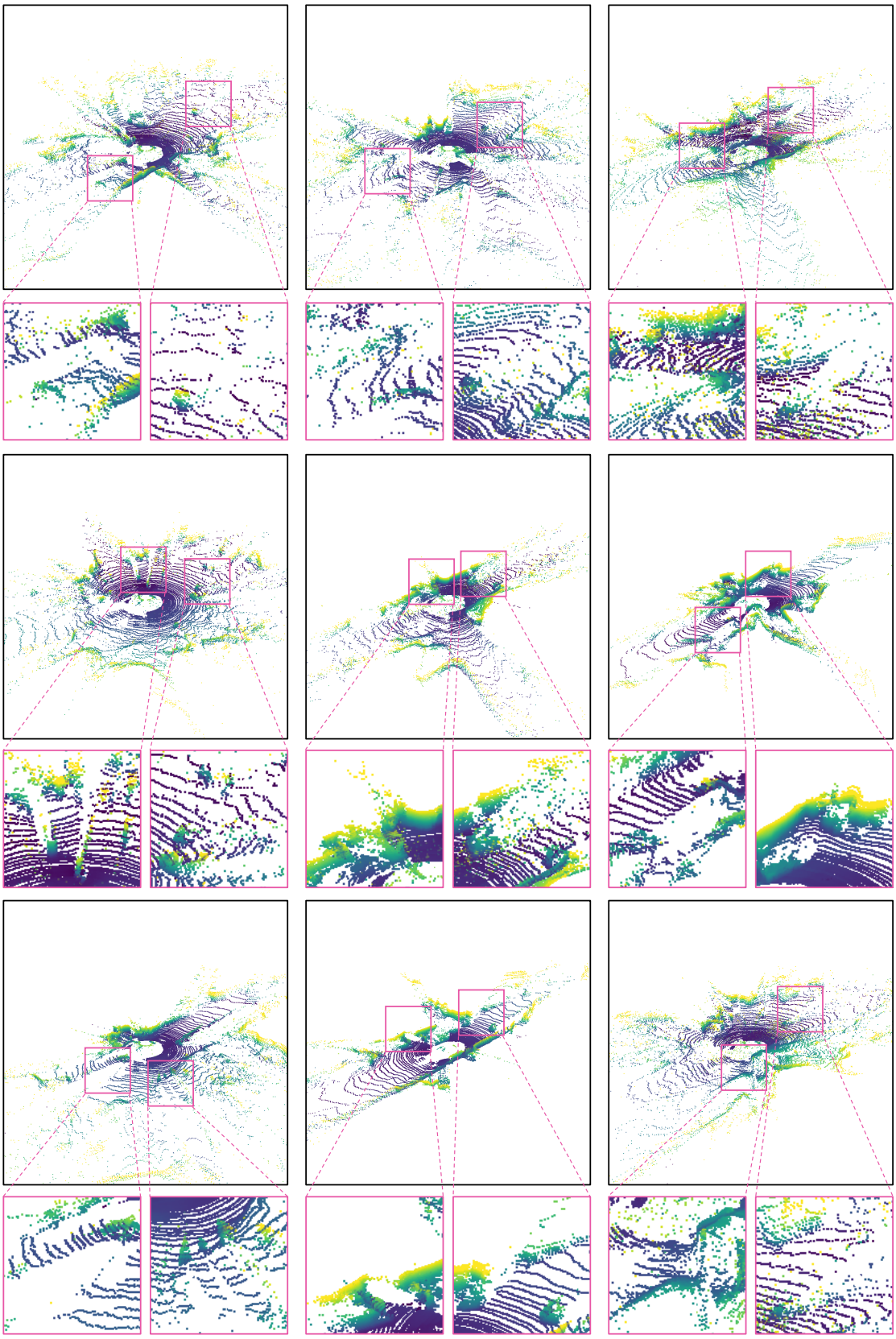

Figure 10. Visualizations of LiDAR point cloud generated under foggy weather conditions.

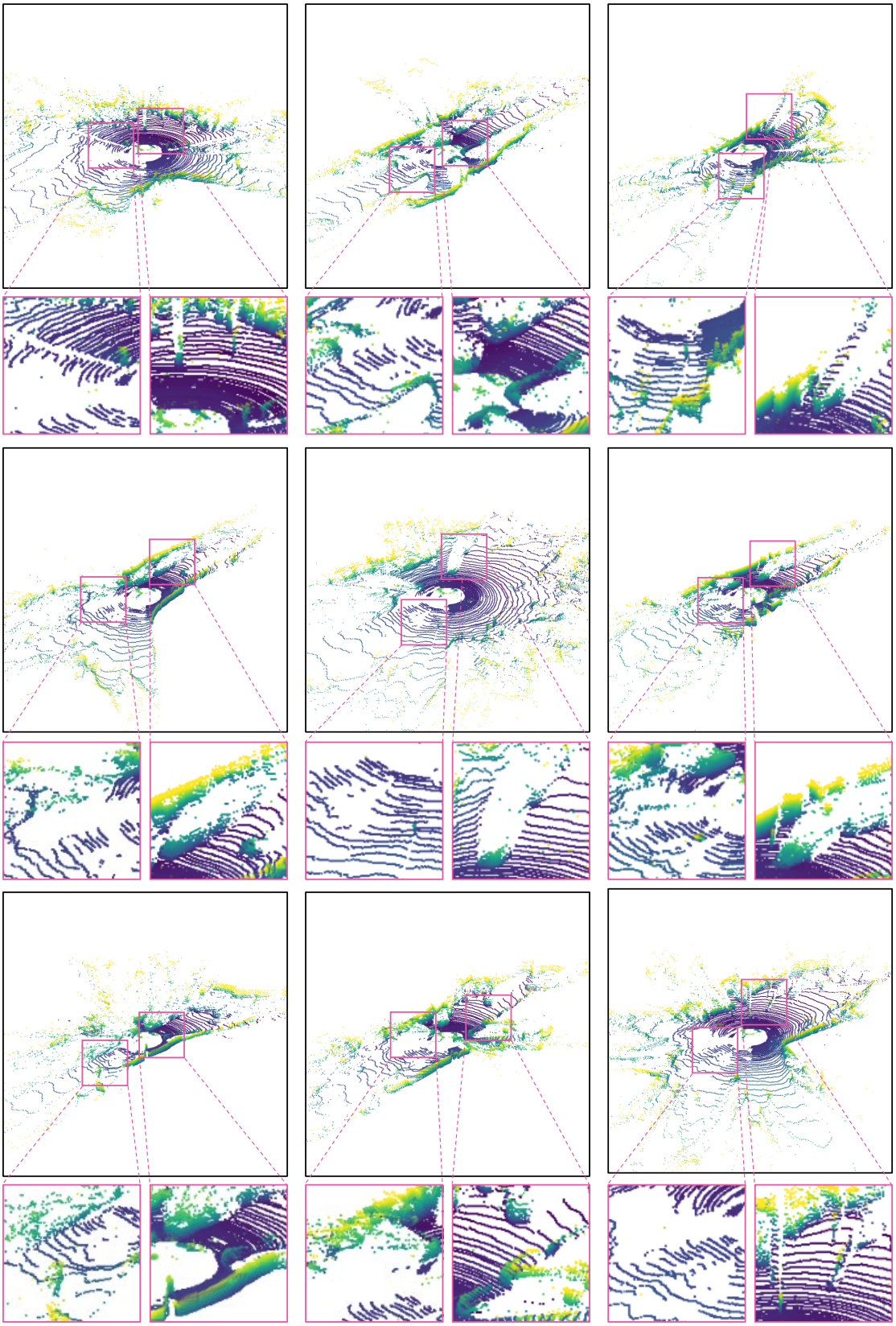

Figure 11. Visualizations of LiDAR point cloud generated under rainy weather conditions.

References

[1] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020. 2

[2] Yuning Chai, Pei Sun, Jiquan Ngiam, Weiyue Wang, Benjamin Caine, Vijay Vasudevan, Xiao Zhang, and Dragomir Anguelov. To the point: Efficient 3d object detection in the range image with graph convolution kernels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2021. 1

[3] Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Fog simulation on real lidar point clouds for 3d object detection in adverse weather. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15283–15292, 2021. 2

[4] Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. Lidar snowfall simulation for robust 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16364–16374, 2022. 2

[5] Velat Kilic, Deepti Hegde, Vishwanath Sindagi, A Brinton Cooper, Mark A Foster, and Vishal M Patel. Lidar light scattering augmentation (lisa): Physics-based simulation of adverse weather conditions for 3d object detection. *arXiv preprint arXiv:2107.07004*, 2021. 2

[6] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. 3

[7] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2

[8] Gregory P Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12677–12686, 2019. 1

[9] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4213–4220. IEEE, 2019. 1, 2

[10] Kazuto Nakashima and Ryo Kurazume. Learning to drop points for lidar scan synthesis. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 222–229. IEEE, 2021. 1

[11] Kazuto Nakashima and Ryo Kurazume. Lidar data synthesis with denoising diffusion probabilistic models. *arXiv preprint arXiv:2309.09256*, 2023. 2

[12] Kazuto Nakashima, Yumi Iwashita, and Ryo Kurazume. Generative range imaging for learning scene priors of 3d lidar data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1256–1266, 2023.

[13] Haoxi Ran, Vitor Guizilini, and Yue Wang. Towards realistic scene generation with lidar diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14738–14748, 2024. 1

[14] Yang Wu, Huihui Song, Bo Liu, Kaihua Zhang, and Dong Liu. Co-salient object detection with uncertainty-aware group exchange-masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19639–19648, 2023. 1

[15] Yang Wu, Kaihua Zhang, Jianjun Qian, Jin Xie, and Jian Yang. Text2lidar: Text-guided lidar point cloud generation via equirectangular transformer. *arXiv preprint arXiv:2407.19628*, 2024. 1, 2

[16] Jing Zhang, Deng-Ping Fan, Yuchao Dai, Saeed Anwar, Fatemeh Saleh, Sadegh Aliakbarian, and Nick Barnes. Uncertainty inspired rgb-d saliency detection. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5761–5779, 2021. 1

[17] Vlas Zyrianov, Xiyue Zhu, and Shenlong Wang. Learning to generate realistic lidar point clouds. In *European Conference on Computer Vision*, pages 17–35. Springer, 2022. 1