LiDAR-to-4DRadar Diffusion Bridge via Cross-Modal Alignment and Translation in Latent Space

Supplementary Material

A. Dataset and Preprocessing

Here, we use Kradar [30] dataset to train and evaluate our model. The Kradar dataset contains 3.5 frames of radar and LiDAR data collected in 58 scenes under various weather, including normal, fog, rain, sleet, overcast, and snow. The original radar data is represented as 4D tensor with the shape of (64, 256, 107, 37) for Doppler, Range, Azimuth, and Elevation. In particular, we removing the first Doppler bin due to its high noise level as shown in the original github code [1]. In addition, for computational efficiency without loss of important information, we crop the radar data along three spatial axes. Finally, the preprocessed radar data has the shape of (63, 192, 86, 32), covering Doppler $[-2, 2], \mathrm{m/s}$, Range $[0, 88.4], \mathrm{m}$, Azimuth $[-48, 48]^\circ$, and Elevation $[-16, 16]^\circ$.

In our experiments, we split the dataset with different settings for synthesis evaluation and downstream tasks evaluation. For synthesis evaluation, we randomly select 1k samples as the test set and other samples as the training set. For downstream tasks evaluation, we follow the split the dataset into three sets: training set, expansion set, and test set, with the ratio of 4:2:1. Specifically, we training our generative model on the training set and synthesize new radar data on the expansion set. Then, we turn to training the downstream tasks model using the synthesized radar data on the expansion set or the combination of the synthesized radar data and the training set. The detailed statistics of the dataset splits are shown in Table 1 and 2.

B. Radar Noise Synthesis

To better capture real-world radar characteristics, we follow prior work [25] to synthesize radar noise and add it to generated radar data at inference. Specifically, the model first generates the Doppler-mean radar volume and the full Doppler vector for key voxels. We then inject synthesized noise into non-key voxels $v_i = (r, a, e)$ around their mean $f_i = \overline{\mathcal{R}}(r, a, e)$. We model the noise with a Gaussian–Softmax distribution:

$$\mathcal{R}_{\mathrm{N}}(v_i) = f_i \cdot \mathrm{Softmax}(\mathbf{z}), \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \delta^2 \mathbf{I}) \tag{10}$$

where $\mathcal{N}(\mathbf{0}, \delta^2 \mathbf{I})$ is a D-dimensional multivariate normal distribution (with D the number of Doppler bins), and δ controls the noise level. Below, we show RA-map examples after noise synthesis with $\delta \in \{1.0, 2.0, 3.0\}$, using the ground-truth Doppler-mean radar map. We set $\delta = 2.0$ as it produces visualizations closest to the ground truth; more refined strategies (e.g., Doppler-bin-specific δ values) are left for future work.

[1] https://github.com/kaist-avelab/K-Radar/blob/main/datasets/kradar_detection_v2_1.py

C. Network Architecture and Training Details

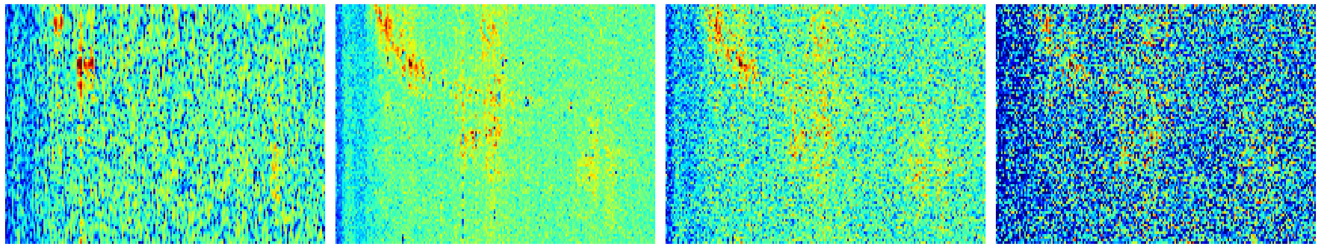
Below we provide the detailed architectures and training configurations for our 4D Radar VAE and 3D U-Net in Table 6 and Table 7, respectively. The LiDAR VAE shares the same backbone as the 4D Radar VAE, but uses a single input/output channel. In particular, L2RLDBb is trained in three sequential stages: (1) train the 4D Radar VAE; (2) freeze the 4D Radar VAE, encode the radar training set into latent representations, and train the LiDAR VAE to align its latent space with the radar latents; (3) pre-encode and store both radar and LiDAR data as latents, then train the latent diffusion bridge on paired LiDAR–radar latents. This staged procedure reduces memory consumption and accelerates training.

The baseline architectures for Latent Diffusion and Pix2pixHD RAE are listed in Table 8 and Table 9, respectively, where we follow their original codes [2] and adopt them into the 3D data. The Latent Pix2pixHD follows the Pix2pixHD RAE design but operates in the VAE latent space (4 channels) and removes the explicit downsampling/upsampling stages.

Most models in our paper are built from 3D ResNet-style blocks, optionally augmented with self- or cross-attention. Specifically, we define the block variants as follows:

- **Res3D**: basic 3D ResNet block.
- **DownRes3D**: 3D ResNet block following with downsampling.
- **UpRes3D**: 3D ResNet block following with upsampling.
- **AttnRes3D**: 3D ResNet block following with self-attention mechanism.
- **DownAttnRes3D**: 3D ResNet block following with self-attention mechanism and downsampling
- **AttnUpRes3D**: 3D ResNet block following with self-attention mechanism and upsampling.
- **CrossAttnDownRes3D**: 3D ResNet block following with cross-attention mechanism and downsampling.
- **CrossAttnUpRes3D**: 3D ResNet block following with cross-attention mechanism and upsampling.

[2] https://github.com/NVIDIA/pix2pixHD/



(a) Original Data (b) $\delta = 1.0$ (c) $\delta = 2.0$ (d) $\delta = 3.0$

Figure 5. The RA map at one Doppler bin with different δ.

Dataset	Total	Normal	Overcast	Rain	Sleet	Fog	Snow
Training	33,994	12,393	802	4,644	6,842	3,483	5,830
Test	1,000	390	22	136	198	95	159

Table 1. Dataset splitting for synthesis evaluation.

Dataset	Total	Object Num	Normal	Overcast	Rain	Sleet	Fog	Snow
Train	20,000	51,102	7,316	455	2,737	4,032	2,007	3,453
Validation	9,992	25,650	3,648	240	1,379	2,005	1,047	1,673
Test	5,000	12,069	1,817	129	664	1,003	524	863

Table 2. Dataset splitting for downstream detection evaluation.

D. Details on Evaluation Metric

In experiments, we report MAE, PSNR, and SSIM in (i) Polar RAE space (Range–Azimuth–Elevation, after Doppler aggregation) and (ii) Cartesian XYZ space (after coordinate projection). Let the generated 4D radar tensor be $\hat{R} \in \mathbb{R}^{D\times R\times A\times E}$ and the ground truth $R \in \mathbb{R}^{D\times R\times A\times E}$, where D = Doppler bins, R = range bins, A = azimuth bins, and E = elevation bins.

Doppler-mean Radar Matrix. We first form Doppler-mean 3D volumes

$$\overline{R}(r, a, e) = \frac{1}{D}\sum_{d=1}^{D} R(d, r, a, e). \tag{11}$$

RAE $\rightarrow$ XYZ projection. Let (r_i, a_j, e_k) be the physical center of bin (i, j, k) with $r_i \in \mathbb{R}_+$, a_j, e_k in radians. We map to Cartesian coordinates:

$$\begin{aligned} x &= r_i \cos(e_k)\sin(a_j), \\ y &= r_i \cos(e_k)\cos(a_j), \\ z &= r_i \sin(e_k). \end{aligned} \tag{12}$$

We construct a fixed Cartesian grid $C \in \mathbb{R}^{X\times Y\times Z}$ covering the spatial extent. Each polar voxel intensity $\overline{R}(r_i, a_j, e_k)$ is splatted (trilinear weighting) into C producing C (ground truth) and $\hat{C}$ (generated). Metrics in XYZ space are then computed between $\hat{C}, C \in \mathbb{R}^{X\times Y\times Z}$.

For clarity, we let (H, W, L) denote (R, A, E) in RAE space and (X, Y, Z) in XYZ space; unless otherwise specified for key-point/Doppler metrics, all evaluations use Doppler-mean volumes.

Mean Absolute Error (MAE). The MAE metric calculates the average absolute difference between the generated radar volume $\hat{R}$ and the ground-truth radar volume R:

$$\text{MAE} = \frac{1}{HWL}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{L}|R(i,j,k) - \hat{R}(i,j,k)|. \tag{13}$$

Peak Signal-to-Noise Ratio (PSNR). The PSNR metric measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. It is defined as:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{MAX^2}{\text{MSE}(\hat{R} - R)}\right), \tag{14}$$

where MAX is the maximum possible pixel range of the radar volume, and MSE is the Mean Squared Error between the generated radar volume $\hat{R}$ and the ground-truth radar volume R.

Structural Similarity Index Measure (SSIM). The SSIM metric evaluates the similarity between two images by considering luminance, contrast, and structural information. It is defined as:

$$\text{SSIM}(R, \hat{R}) = \frac{(2\mu_R\mu_{\hat{R}} + C_1)(2\sigma_{R\hat{R}} + C_2)}{(\mu_R^2 + \mu_{\hat{R}}^2 + C_1)(\sigma_R^2 + \sigma_{\hat{R}}^2 + C_2)}, \tag{15}$$

Steps	Time↓	XYZ Space Metric			RAE Space Metric			Key Point/Doppler Metric		
		MAE↓	PSNR↑	SSIM↑	MAE↓	PSNR↑	SSIM↑	MAE↓	PSNR↑	IoU↑
Steps =1	0.1004	0.1028	27.51	0.8740	0.1626	25.37	0.6718	0.6078	22.39	0.1959
Steps =3	0.2385	**0.05255**	33.01	**0.9092**	**0.09820**	**29.21**	**0.7324**	**0.5452**	**23.44**	**0.2885**
Steps =5	0.4139	0.05308	**33.11**	0.9061	0.09901	29.10	0.7274	0.5425	23.49	0.2846
Steps =10	0.8487	0.05339	33.02	0.9043	0.09976	29.01	0.7244	0.5442	23.46	0.2794
Steps =20	1.0540	0.05358	32.97	0.9036	0.1002	28.97	0.7230	0.5451	23.44	0.2768

Table 3. The generative performance under different diffusion steps. Best value in each column is highlighted in bold.

λ_{bce}	XYZ Space Metric			RAE Space Metric			Key Point/Doppler Metric		
	MAE↓	PSNR↑	SSIM↑	MAE↓	PSNR↑	SSIM↑	MAE↓	PSNR↑	IoU↑
0.0	0.03192	40.62	0.9781	0.06092	34.61	0.8649	0.4629	25.04	0.08655
0.3	0.03192	40.62	0.9781	0.05829	35.10	**0.8688**	0.4610	25.07	0.8145
0.5	0.03063	42.01	**0.9791**	**0.05817**	**35.11**	0.8685	**0.4607**	**25.10**	0.8798
0.7	**0.03040**	**42.50**	0.9790	0.05919	34.94	0.8653	0.4608	25.06	**0.9219**

Table 4. The reconstruction performance of 4DRadar VAE with different λ_{bce} values. The best value in each column is highlighted in bold.

λ_{cont}	XYZ Space Metric			RAE Space Metric			Key Point/Doppler Metric		
	MAE↓	PSNR↑	SSIM↑	MAE↓	PSNR↑	SSIM↑	MAE↓	PSNR↑	IoU↑
0.0	0.05332	32.85	0.9037	0.09920	29.00	0.7236	0.5517	23.35	0.2626
0.3	**0.05250**	**33.01**	**0.9092**	0.09820	**29.21**	0.7324	**0.5452**	23.44	0.2885
0.5	0.05280	32.98	0.9089	**0.09800**	29.11	0.7313	0.5484	23.43	0.2765
0.7	0.05251	33.97	0.9090	0.09827	29.14	**0.7325**	0.5463	**23.45**	**0.2905**

Table 5. The generative performance under different λ_{cont} values. The best value in each column is highlighted in bold.

where μ_R and $\mu_{\hat{R}}$ are the mean values of R and $\hat{R}$, respectively; σ^2_R and $\sigma^2_{\hat{R}}$ are the variances of R and $\hat{R}$, respectively; $\sigma_{R\hat{R}}$ is the covariance between R and $\hat{R}$; and C_1 and C_2 are small constants to stabilize the division.

E. Discussion on Inference Steps

Here, we analyze the influence of different diffusion steps during inference on the generative performance and time consumption. Tab. 3 shows the quantitative results. We can find that using 3 steps during inference achieves the best generative performance with a moderate time consumption.

F. Parameter Analysis on λ_{bce} and λ_{cont}

Here, we analyze the impact of two loss weights in the 4D Radar VAE and LiDAR VAE: λ_{bce} for the BCE key-voxel mask loss $\mathrm{BCE}(M_{\mathrm{key}}, \hat{M}_{\mathrm{key}})$ and λ_{cont} for the contrastive loss $\mathcal{L}_{cont}$. Tab. 4 reports the reconstruction performance of the 4D Radar VAE under varying λ_{bce}. The model performs best at $\lambda_{bce} \in \{0.5, 0.7\}$; we set $\lambda_{bce} = 0.5$ to balance the three types of metrics. Tab. 5 shows the generative performance of L2RLDB with LiDAR VAEs under different λ_{cont} values. Performance is similar for $\lambda_{cont} > 0$ and consistently better than $\lambda_{cont} = 0$, demonstrating the effectiveness of the contrastive loss. We set $\lambda_{cont} = 0.3$ to achieve the best overall metrics.

G. More Qualitative Results

Here, we provide additional qualitative results of L2RLDB and the baselines. First, we compare RA maps across weather conditions. Doppler-mean RA maps are shown in Fig. 9 and Fig. 10, while single-Doppler-bin RA maps and key-voxel comparisons are given in Fig. 11–14. Overall, our method produces structures visually closest to the ground truth, better preserving target returns and background clutter while avoiding over-smoothing and speckle artifacts. By contrast, baselines tend to blur weak echoes or introduce spurious speckles. Second, we show AE (Azimuth–Elevation) maps of the generations in Fig. 6 and Fig. 7. Our results exhibit the similar highlighted region with the ground truth, consistent with the RA-view observations. Third, consistent with Sec. 5.2, synthetic radar under fog and sleet yields limited or no improvement in downstream detection. As shown in Fig. 8, severe attenuation and low SNR suppress weak echoes and reduce local contrast, making small objects hard to separate from background clutter, which causing the high error for key-voxel identification in the last line.

Component	Parameter	Value
Encoder	Input Channels	64 (1 for Doppler-mean RAE + 63 for Doppler vectors)
	Input Resolution	[192,96,32]
	Block Types	[DownRes3D, DownRes3D, DownAttnRes3D]
	Block Output Channels	[64, 128, 256]
	Attention Header Channels	8
	Layers per Block	2
Latent Space	Mid Block	AttnRes3D
	Layers per Block	2
	Output	μ, $\log(\sigma^2)$
	Latent Shape	$(4, 48, 24, 8)$
	Downsampling Factor	$4\times$
Decoder	Block Types	[AttnUpDecoder3D, UpDecoder3D, UpDecoder3D]
	Block Output Channels	[256, 128, 64]
	Attention Header Channels	8
	Layers per Block	2
	Output Channels	64 (1 for Doppler-mean RAE + 63 for Doppler vectors)
	Key-Voxel Mask Output Channel	1
Training	Batch Size	16
	Learning Rate	1×10^{-4}
	Optimizer	Adam
	Adam β_1 / β_2	0.9 / 0.999
	Weight Decay	1×10^{-6}
	Total Parameters	$\sim$34.77M
	Iteartions	100k

Table 6. 4DRadar VAE network architecture and training configuration

Component	Parameter	Value
Down Blocks	Input Channels	4
	Input Resolution	$(48, 24, 8)$
	Block Types	[DownRes3D, DownRes3D, AttnDownRes3D]
	Block Output Channels	[128, 256, 512]
	Attention Head Channels	32
	Layers per Block	2
Mid Block	Block Type	AttenRes3D
	Layers per Block	2
Up Blocks	Block Types	[AttnUpRes3D, AttnUpRes3D, UpRes3D]
	Skip Connections	True
	Block Output Channels	[512, 256, 128]
	Attention Head Channels	32
	Layers per Block	2
	Output Channels	4
Time Step	Diffusion steps	1000
	Scheduler	Linear
	Enbedding Types	Sinusoidal
	Time Scale Shift	AdaLN
Training	Batch Size	64
	Learning Rate	1×10^{-4}
	Optimizer	AdamW
	Adam β_1 / β_2	0.95 / 0.999
	Weight Decay	1×10^{-6}
	Total Parameters	$\sim$162.10M
	Iteartions	100k

Table 7. The UNet3D network architecture in L2RLDB and training configuration

Component	Parameter	Value
Down Blocks	Input Channels	4
	Input Resolution	$(48, 24, 8)$
	Block Types	[DownRes3D, CrossAttnDownRes3D, CrossAttnDownRes3D]
	Block Output Channels	[128, 256, 512]
	Attention Head Channels	32
	Layers per Block	2
Mid Block	Block Type	AttenRes3D
	Layers per Block	2
Up Blocks	Block Types	[CrossAttnUpRes3D, CrossAttnUpRes3D, UpRes3D]
	Skip Connections	True
	Block Output Channels	[512, 256, 128]
	Attention Head Channels	32
	Layers per Block	2
	Output Channels	4
Time Step	Diffusion steps	1000
	Scheduler	Linear
	Enbedding Types	Sinusoidal
	Time Scale Shift	AdaLN
Training	Batch Size	64
	Learning Rate	1×10^{-4}
	Optimizer	AdamW
	Adam β_1 / β_2	0.95 / 0.999
	Weight Decay	1×10^{-6}
	Total Parameters	$\sim$178.69M
	Iteartions	100k

Table 8. The UNet3D network architecture in latent diffusion and training configuration

Component	Parameter	Value
Gneration	Input Channels	1 (LiDAR)
	Input Resolution	(192, 96, 32)
	Down Blocks	[Conv3D, Conv3D, Conv3D, Conv3D]
	Down Stride	[1, 2, 2, 2]
	Down Channels	[64, 64, 126, 256]
	Down Kernel Size	[7, 3, 3, 3]
	Resnet Blcoks	[Resnet] *6
	Resent Channels	[256] * 6
	Up Blocks	[ConvT3D, ConvT3D, ConvT3D, Conv3D]
	Up Stride	[2, 2, 2, 1]
	Up Channels	[256, 126, 64, 64]
	Up Kernel Size	[3, 3, 3, 7]
	Activiation Function	ReLU
	Output Channel	64 (Radar)
Discriminator	Input Channels	64 (Radar)+1 (LiDAR)
	Input Resolution	(192, 96, 32)
	Layers	[Conv3D, Conv3D, Conv3D, Conv3d]
	Layers Stride	[2, 2, 1, 1]
	Layers Channels	[64, 128, 256, 1]
	Up Kernel Size	[4, 4, 4, 4]
	Activiation Function	LeakyReLU(0.2)
	Output Channel	1
Training	Batch Size	32
	Learning Rate	1×10^{-4}
	Optimizer	Adam
	Adam β_1 / β_2	0.95 / 0.999
	Weight Decay	1×10^{-6}
	Total Parameters	$\sim$95.34M
	Iteartions	30k

Table 9. The network architecture in Pix2pixHD RAE and training configuration

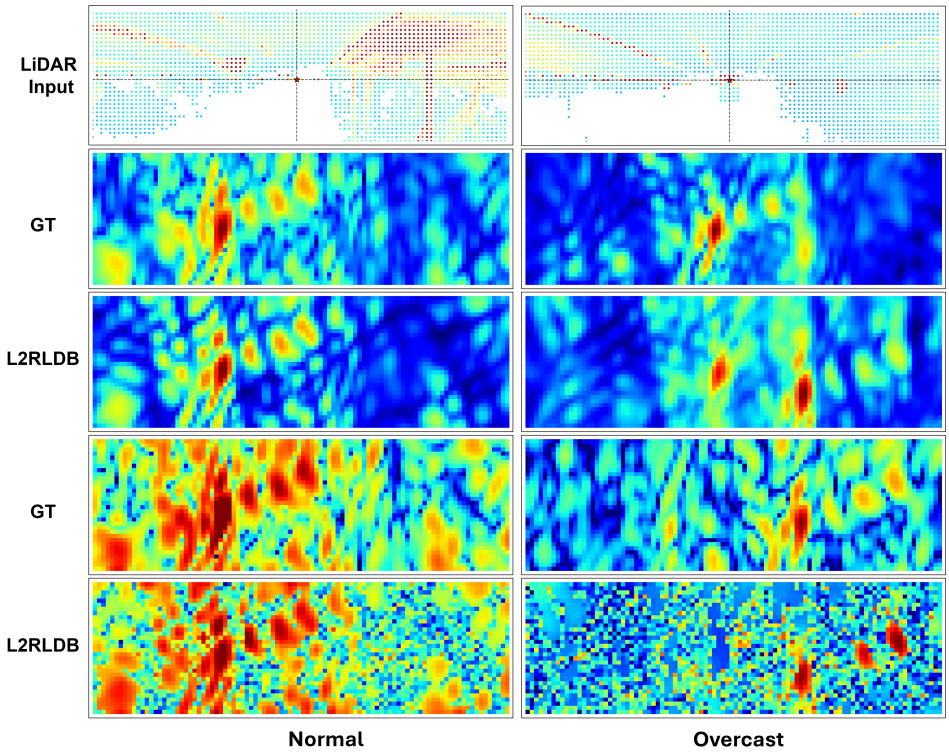


Figure 6. Azimuth–Elevation (AE) maps at a fixed range bin. Top row: ground truth. Subsequent rows show (i) Doppler-averaged AE maps for the ground truth and L2RLDB, and (ii) AE maps at a single Doppler bin. Weather conditions are labeled below each column.

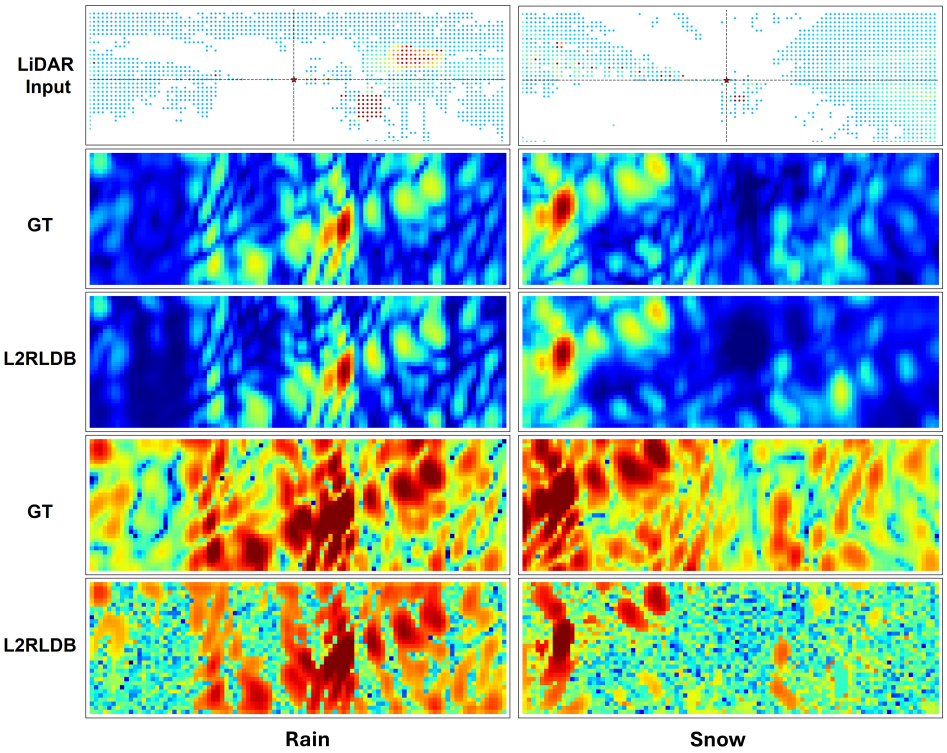


Figure 7. Azimuth–Elevation (AE) maps at a fixed range bin. Top row: ground truth. Subsequent rows show (i) Doppler-averaged AE maps for the ground truth and L2RLDB, and (ii) AE maps at a single Doppler bin. Weather conditions are labeled below each column.

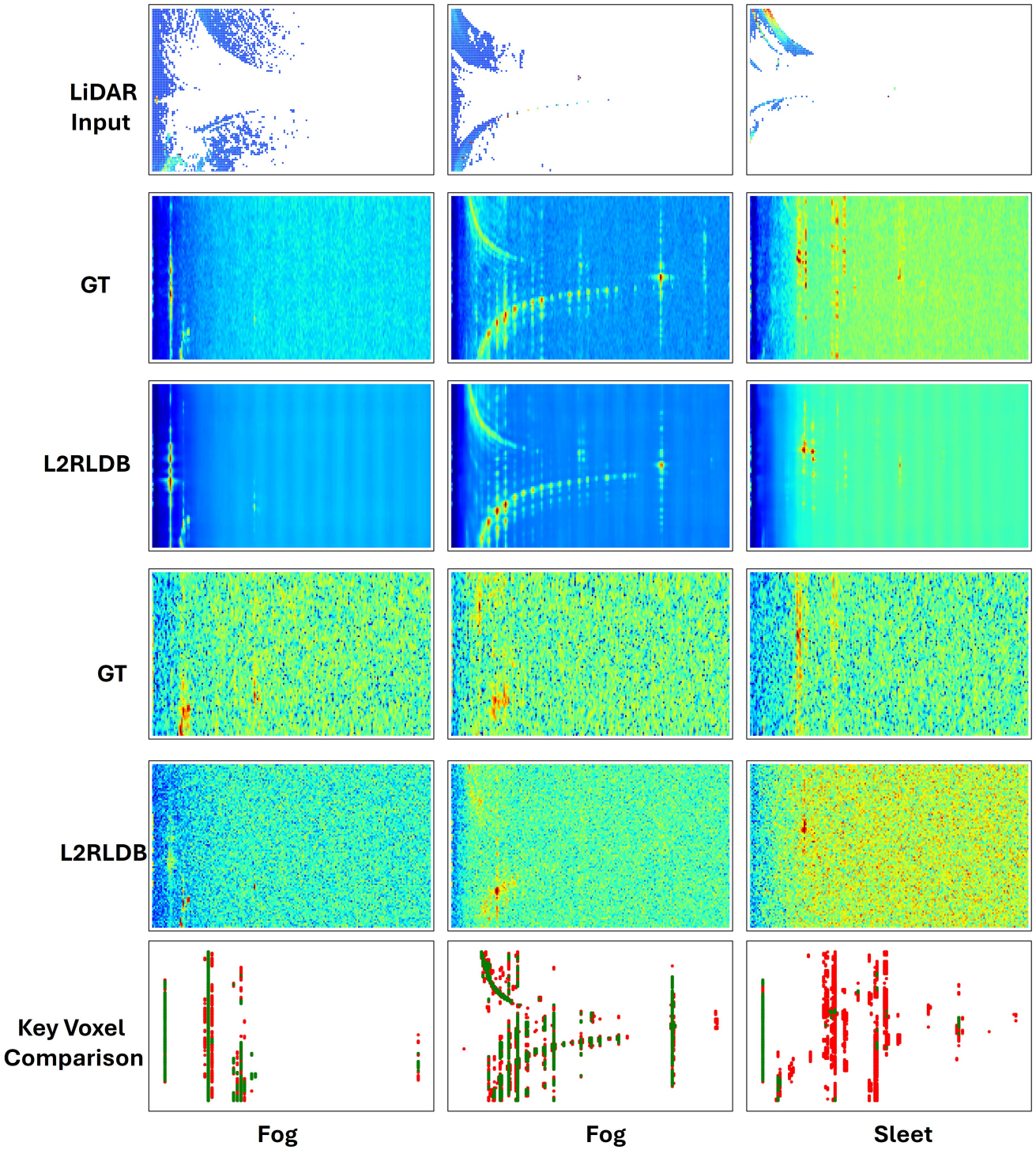


Figure 8. Comparison between ground-truth 4D radar and L2RLDB-generated results in fog and sleet conditions. Top row: LiDAR input. Middle four rows: Doppler-mean RA maps and single-Doppler-bin RA maps for the ground truth and L2RLDB. Bottom row: key-voxel identification results.

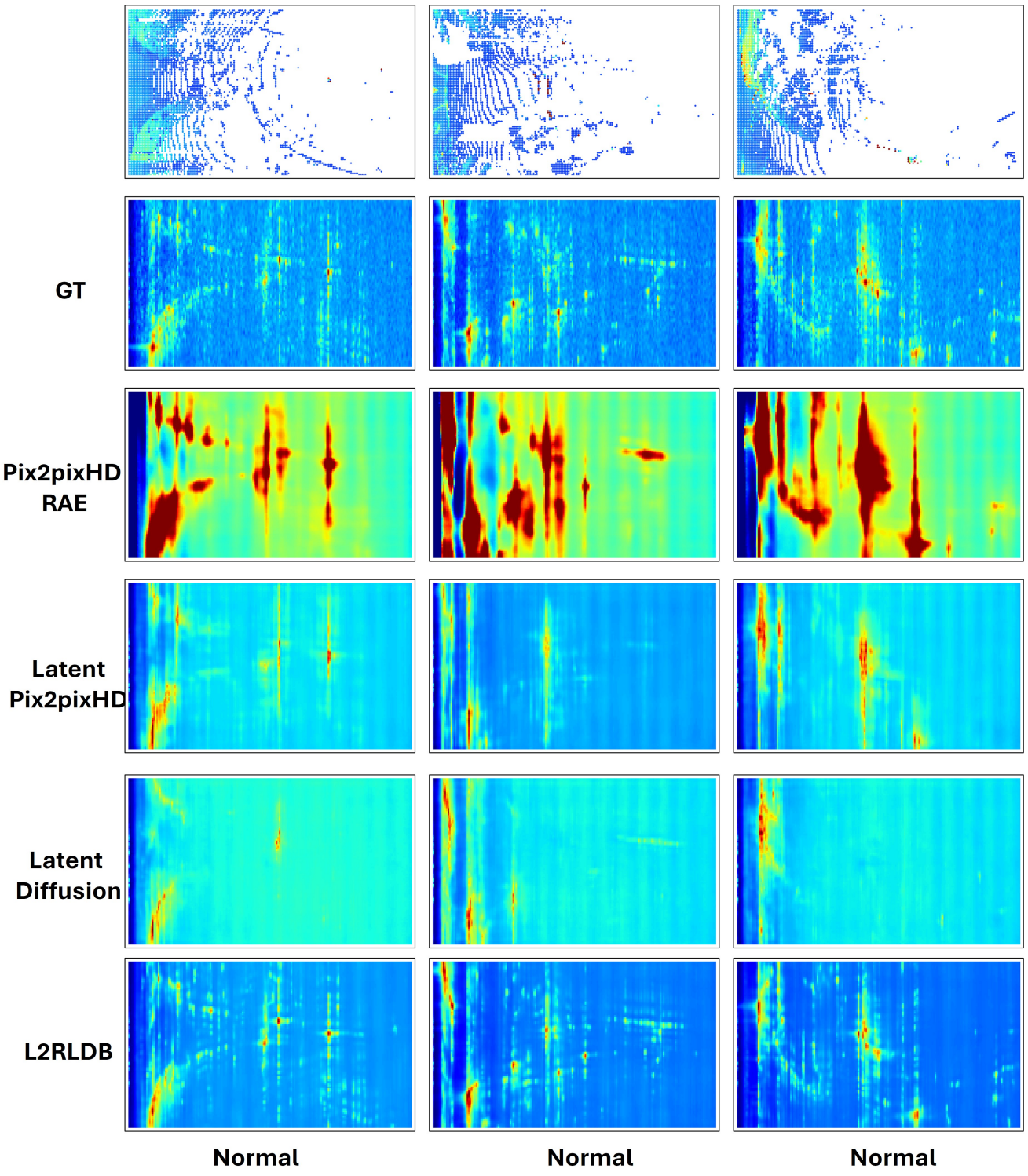


Figure 9. More results of Doppler-mean RA maps under different weather conditions. Top row: ground truth. Subsequent rows show results from baselines and L2RLDB. Weather conditions are labeled below each column.

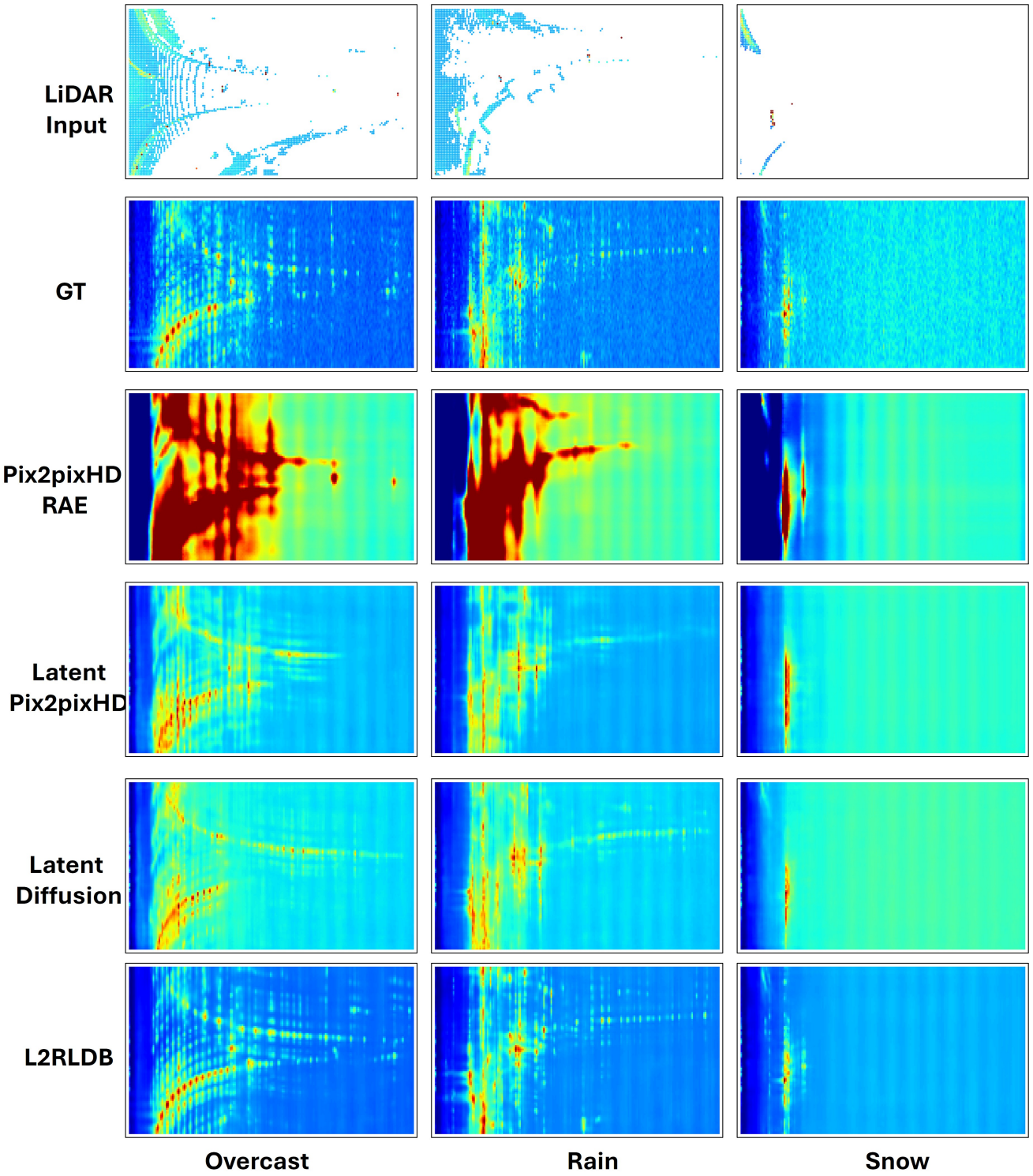


Figure 10. More results of Doppler-mean RA maps under different weather conditions. Top row: ground truth. Subsequent rows show results from baselines and L2RLDB. Weather conditions are labeled below each column.

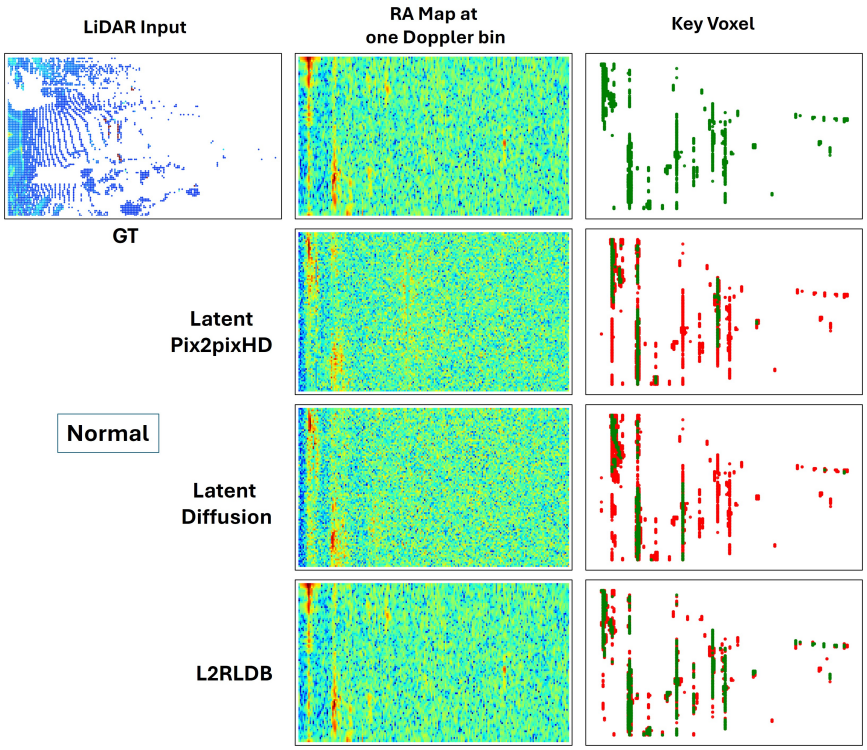


Figure 11. More RA maps at a single Doppler bin under the normal weather. The top row shows ground truth; subsequent rows show baselines and L2RLDB.

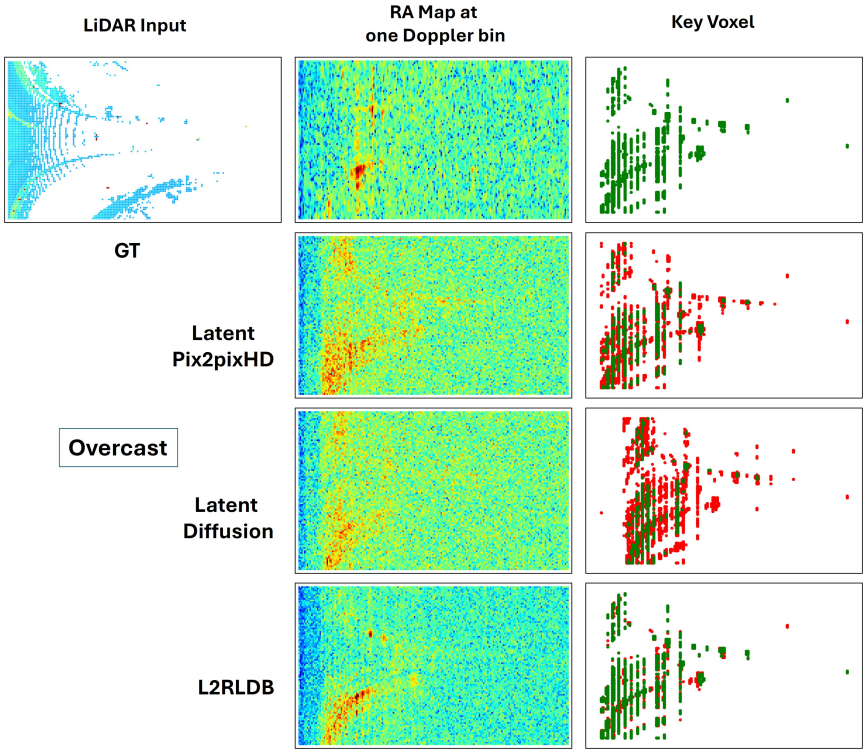


Figure 12. More RA maps at a single Doppler bin under the overcast weather. The top row shows ground truth; subsequent rows show baselines and L2RLDB.

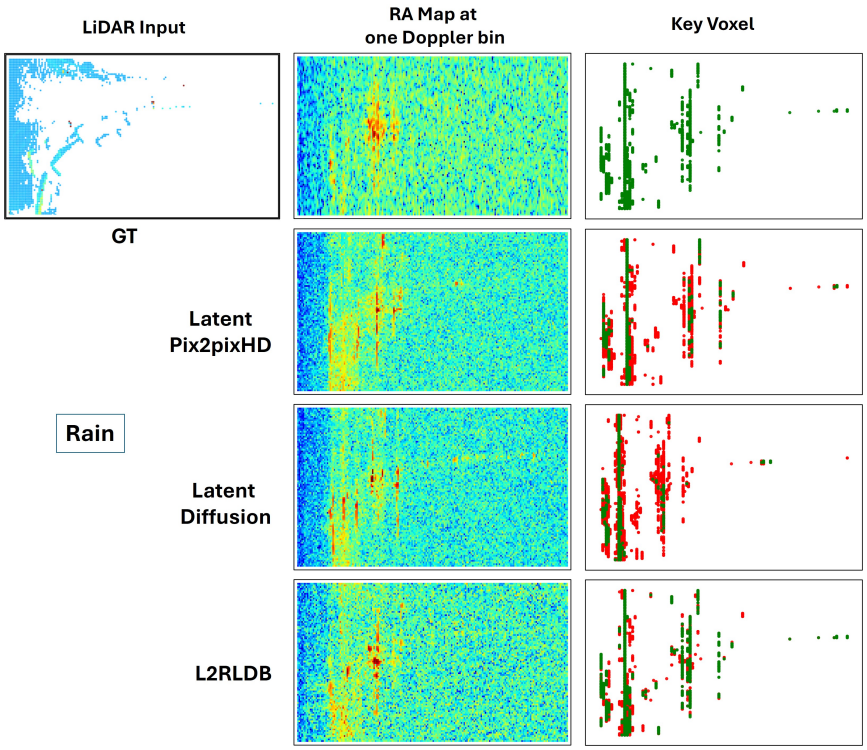


Figure 13. More RA maps at a single Doppler bin under the rain weather. The top row shows ground truth; subsequent rows show baselines and L2RLDB.

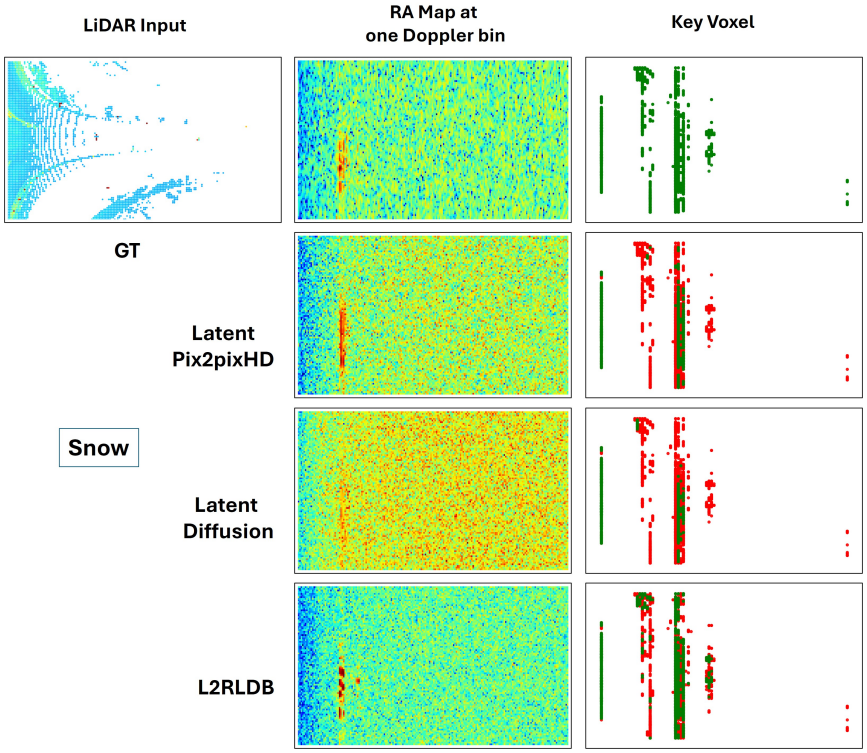


Figure 14. More RA maps at a single Doppler bin under the snow weather. The top row shows ground truth; subsequent rows show baselines and L2RLDB.