Multi-Sensor Alignment for Weather Simulations

Samsad Alam, Devyani Lambhate, Aditya Mohan, Vishal Kumar, and Vaibhav Katewa

***Abstract*—Perception tasks for autonomous vehicles need to work satisfactorily in adverse weather conditions. Due to lack of real-world weather datasets, weather simulations are a promising alternative. To ensure simulations closely mirror real-world weather data, it's crucial that they represent the same weather characteristics, including severity and particle positioning, across different sensors. To achieve this, we propose the Reference Dataset Alignment Method (ReDAM) for weather intensity alignment in fog and Unified-weather-edit (inspired by Weather-edit [1]) for particle positioning alignment in rain and snow. We validate both alignment methods using statistical and geometrical tests, respectively. We find that 3D detection models for non-aligned versions tend to be overly optimistic as compared to aligned versions. We also show the aligned-multi-sensor simulation's effectiveness for achieving robustness for 3D object detection task by finetuning existing sensor fusion models on it.**

I. Introduction

In autonomous driving, 3D perception aims to understand the surrounding environment in full 3D using multiple sensors such as LiDAR and Camera. A key task is 3D object detection, which predicts 3D bounding boxes for objects like vehicles, pedestrians, and cyclists. For reliable deployment, such systems must operate robustly under adverse weather conditions, including fog, rain, and snow. However, widely used datasets contain only a limited number of adverse weather samples, leading to poor generalization in such scenarios.

To mitigate data scarcity, prior works [2] employ simulation-based approaches to generate adverse weather data. While effective in increasing representation of adverse weather conditions in the dataset, these methods often simulate each sensor independently, resulting in misaligned representations of weather characteristics such as severity and particle distribution across modalities. This inconsistency deviates from real-world conditions. In this work, we propose sensor-aligned weather simulation methods to ensure alignment in weather severity and particle position.

Some of the previous works have proposed weather simulated datasets. Nuscenes-C [3] is a commonly used corrupted dataset that uses physics based weather simulation methods for LiDAR and image enhancements from image processing or Computer-Vision models to simulate visually realistic weather Camera Images, but it does not address sensor alignment. MSC-Bench [4] also introduces weather simulations using a snow simulator [5] and a fog simulator [6] for LiDAR and then matches weather severity in Camera using snowfall levels and fog parameters. Weather-edit [1] proposes aligning particle positions in different views and temporal frames of Camera Images, but does not handle cross-sensor alignment. Although some of the approaches focus on weather severity alignment, to the best of our knowledge, there are no existing approaches to address weather particle positioning alignment across sensors.

Each weather type requires a tailored sensor alignment method. Homogeneous weather mediums like fog or dust, where a large number of weather particles are uniformly distributed in the 3D space requires only intensity alignment. In contrast, non-homogeneous mediums like snow or rain, where weather particles are much bigger and sparser, require position alignment of weather particles along with intensity based alignment. For example, a rain streak at (x, y, z) positions in 3D space affects both Camera and LiDAR rays passing through it. To achieve this, we propose Reference Dataset based Alignment Method (ReDAM) for homogeneous weather mediums and Unified-weather-edit (based on Weather-edit [1]) for non-homogeneous mediums. Our key contributions are listed below.

1) *Re*ference *D*ataset based *A*lignment *M*ethod (ReDAM) which aligns the fog severity in simulated LiDAR and Camera using a real-world weather dataset as reference.
2) Unified-weather-edit model to achieve (a) position alignment of weather particles in simulated LiDAR and Camera. This alignment uses a common 3D world coordinate representation for particles. (b) Temporal alignment of particles in 3D world coordinates, which results in temporal alignment in both sensors.
3) Alignment verification using KS-Statistics [7] and image quality assessment in ReDAM; Geometric verification of particle alignment in Unified-weather-edit with re-projection error, and perceptual verification with sensor fusion based 3D object detection.

II. Weather Simulation and Sensor Alignment

In this section, we first brief about our weather simulation methodology and then present sensor alignment methods.

A. Weather Simulation

LiDAR weather simulation: We use physics-based models for LiDAR simulations as they provide parametric controllability of weather severity. To capture the effect of atmospheric media on LiDAR reflections, we leverage the LiDAR impulse response given by [8]

$$P_{rec}(R) = C_A \int_0^{2R/c} P_{tr}(t) H\left(R - \frac{ct}{2}\right) dt, \qquad (1)$$

All authors are with the Department of Cyber-Physical Systems at the Indian Institute of Science, Bengaluru. Email IDs: samsadalam@iisc.ac.in, devyani.lambhate@fsid-iisc.in, adityamohan@iisc.ac.in, vishal.kumar1@fsid-iisc.in, and vkatewa@iisc.ac.in.

This work is supported by Sony India via the Faculty Innovation Award.

(a) Clean LiDAR BeV (b) Simulated LiDAR BeV (c) Clean Image (d) Simulated Snow Image

Fig. 1. Simulation of snowy weather on a LiDAR sample ((a)-(b)) and a Camera image sample ((c)-(d)). Red points in (b) appear due to backscatter from snowflakes. There is also a loss of points far away from LiDAR, noticeable by comparing gray points in (a) and (b). (d) contains both the background edit effect (snow deposit on the footpath) and rendered snowflake particles.

where $P_{rec}(\cdot)$ is the received power, R is the distance from LiDAR, $P_{tr}(\cdot)$ is the transmitted power, $H(\cdot)$ is the channel response, C_A is the sensor constant and c is the speed of light.

We model the weather effects by selecting appropriate $H(R)$ for fog, rain, and snow. Target object response is always modeled using impulse function. Fog response is modeled in [9] as a homogeneous medium with the step function response characterized by attenuation factor α and fog reflectivity β. Snow and rain are modeled by [10] and [11], respectively as discrete media with particles populating the field of view of LiDAR. Each snow/rain particle is characterized by an occlusion ratio signifying portion of beam that particle blocks and reflectivity of the particle. Each particle's response is characterized by impulse function just like target object. The received power is computed by superposing response from each particle in the beam. Upon computation of received power in a beam we detect where the peak occurs and that becomes our modified location and the peak power becomes modified intensity. Figure 1 (a) and (b) show the snowy weather simulation on a sample point cloud.

***Camera weather simulation**:* To simulate weather in Camera images, we follow methods described in TSIT [12] and Weather-edit.

TSIT is a style-transfer-based image-to-image translation framework to synthesize images under a desired visual condition by translating base images into a target style domain. In this framework, the base image is treated as the *content image*, providing the semantic structure and spatial layout of the scene, while images from the target domain serve as *style references* that capture the corresponding visual characteristics of the desired condition.

Weather-edit is another model designed to simulate weather effects across multiple views and frames, while maintaining consistency in both multi-view and temporal (multi-frame) dimensions. This is achieved through a two-stage process.

(a) The first step is **background editing**, where a diffusion model is used to generate weather-specific background effects, such as, whitish snowy or cloudy rainy background.

(b) The next step is **particle rendering**, which models the weather particles with 4D Gaussian field. Each weather type is represented by Gaussian particles with attributes $A_i = \{C_i, P_i, R_i, S_i, O_i\}$ which represent color, position, rotation, scale, and opacity of Gaussian i, respectively, chosen to mimic the real weather properties of snowflakes or raindrops. Since the weather particles are simulated in a unified 3D world coordinate system and then shared across all views, this inherently ensures multi-view consistency. Further, the motion of the particles in 3D world coordinate is approximated using a constant terminal velocity vector $\mathbf{D}$, which is used to update the position of each particle as: $\mathbf{P}_{world}(t + \Delta t) = \mathbf{P}_{world}(t) + \mathbf{D}\Delta t$. This results in temporal consistency in particles across frames. Figure 1 (c) and (d) show the snow weather simulation on a Camera image sample.

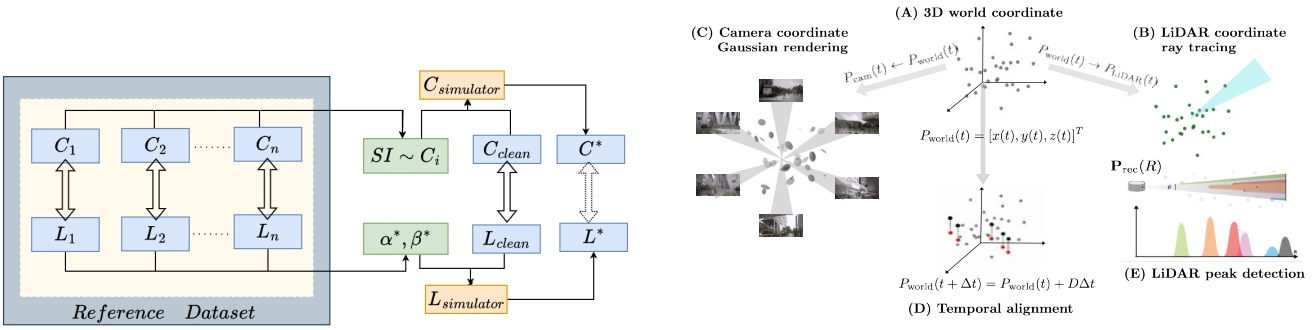
B. Weather Alignment Across Sensors

We propose two different methods of alignment between LiDAR and Camera: **ReDAM for weather severity alignment** that aligns weather severity for homogeneous weather medium like fog, and **Position-based particle alignment using Unified-weather-edit** that aligns particle positions for non homogeneous medium like snow and rain.

*1) **ReDAM for Fog**:* To achieve fog severity alignment between simulated LiDAR and Camera, we propose to leverage a reference dataset that has real and aligned (LiDAR-Camera) fog data. We choose *Seeing Through Fog* (STF) [13] dataset for this purpose. ReDAM consists of two steps - (i) Aligning the simulated LiDAR weather with the STF LiDAR data (ii) Aligning the simulated Camera weather with the STF Camera data.

Since the STF dataset contains aligned LiDAR and Camera observations under real fog, aligning our simulations to STF implicitly produces aligned weather simulations in both sensors. Next, we explain the above two steps in detail.

Severity Alignment of Simulated LiDAR with STF LiDAR: Recall that we use the physics-based model (1) for LiDAR fog simulation. In this model, the fog severity is captured by parameters α and β. Thus, we propose to tune these parameters to minimize the **KS-Statistics** [7] between the intensity distributions of simulated and STF point cloud. We estimate the empirical cumulative distribution function (ECDF) $F_{real}(x)$



(1) ReDAM alignment

(2) Unified-weather-edit alignment

Fig. 2. (1) **ReDAM**: $\{C_1, \cdots, C_n\}$ and $\{L_1, \cdots, L_n\}$ represent Camera and LiDAR data from the reference dataset. SI is a Sample Image from Camera data. LiDAR and Camera simulators $(C_{simulator}, L_{simulator})$ take the clean samples (C_{clean}, L_{clean}) and generate the weather simulated samples (C^*, L^*). (2) **Unified-weather-edit**: Particle positions in 3D world coordinate (**A**) are transformed to LiDAR coordinate (**B**) and Camera coordinate (**C**). Transformed coordinates are used for particle simulation in each sensor.

of intensity X of STF point cloud and $F_{sim,(\alpha,\beta)}(x)$ of simulated point cloud from respective samples, and compute the parameter values as:

$$(\alpha^*, \beta^*) = \underset{(\alpha,\beta)}{\operatorname{argmin}} \left[D_{KS} = \sup_x |F_{sim,(\alpha,\beta)}(x) - F_{real}(x)| \right]$$

We then use the tuned parameters α^*, β^* to simulate fog on our desired dataset as mentioned in Section II-A.

Severity Alignment of Simulated Camera with STF Camera: We employ TSIT for fog simulation in camera images, where a target style image encodes the desired weather conditions and severity. For each image from the nuScenes-mini [14]dataset, a corresponding target style image is randomly sampled from the STF dataset at the same fog level and passed through the TSIT architecture to generate the simulated image. This procedure ensures that the severity of the fog in the simulated image and the STF dataset is closely aligned. The overall ReDAM process is illustrated in Figure 2-(1). To generate simulations with different severities, the ReDAM process can be repeated for real-weather reference subsets representing different weather severities.

*2) **Unified-weather-edit for Snow and Rain**:* To align particle (snowflake/raindrop) positions between LiDAR and Camera, we propose to use a unified 3D World coordinate framework similar to that used in Weather-edit. Since LiDAR is largely uncorrelated with image background changes [15], we adapt only the particle position alignment step described in Subsection II-A. We first determine the number of particles to be generated based on the snowfall/rainfall rate and the terminal velocity. Then, we render each particle in the 3D World coordinate by randomly sampling its coordinates $[x, y, z]$ uniformly in the field of view of sensors (refer to Figure 2, 2-A). Next, we transform these particle positions in LiDAR and Camera coordinates with the following transformation equation

$$\mathbf{P}_s = \mathbf{W}_{world \rightarrow s} \mathbf{P}_{world},$$

where $s \in [LiDAR, Camera]$, and $\mathbf{W}_{world \rightarrow s}$ is $\mathbb{R}^{3\times3}$ the World to LiDAR/Camera transformation matrix, respectively. Since the LiDAR and Camera particle positions are derived from a unified world coordinate, this ensures spatial particle alignment between LiDAR and Camera. Further, the temporal consistency of the world coordinates intrinsically results in temporal alignment in each of Camera and LiDAR. These positions are then utilized independently in the respective sensor to simulate particle effects, as explained next.

LiDAR simulation: We place the particles at location $\mathbf{P}_{LiDAR}$. The diameter of the particles is chosen as described in Subsection II-A. We find particles in each LiDAR beam and compute occlusion ratio for each particle intersecting the beam. Then, we compute the response from each particle as described in Subsection II-A to simulate the weather.

Camera simulation: We place Gaussian means at the particle locations $\mathbf{P}_{cam}$ as illustrated in Figure 2, (2-C). Then, we select associated attributes like scale, rotation, opacity, and color. These particle Gaussians are subsequently integrated with scene Gaussians from weather-edit and rendered jointly to generate images containing simulated weather particles.

III. Experimental Setup and Results

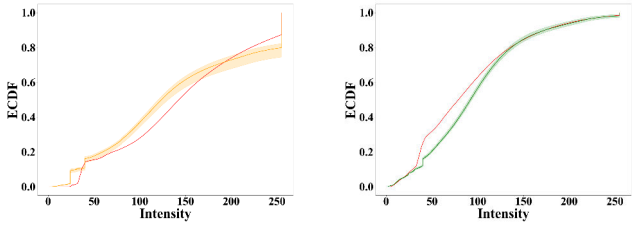
For the ReDAM method, parameter tuning of α and β, and style image sampling are done using the STF dataset, comprising 2183 clean samples and fog splits with 1049 light and 881 dense fog samples. For particle alignment, we use the nuScenes-mini dataset, as STF's limited views are insufficient for reliable 3D reconstruction in Weather-edit. To reduce the computational cost of weather simulation and 3D reconstruction in Weather-edit, we adopt the nuScenes-mini subset instead of full trainval set. The nuScenes-mini subset retains full sensor configuration but reduced dataset size (31,206 frames, 404 annotated samples, and 18,538 object annotations from 10 scenes across 23 classes). All sensor fusion evaluations are also conducted on nuScenes-mini.

Simulation parameters

The parameters α, β are tuned with grid search over range $\alpha \in \mathcal{A} = \{0.001, 0.005, 0.01, 0.02, 0.04, 0.06, 0.08, 0.1\}$ and $\beta \in \mathcal{B} = 0.023 \times \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}/MOR$ where $MOR = 3/\alpha$ defines visibility range. For particle alignment in snow, we have chosen snowfall rate $2.0mm/hr$ and terminal velocity $\mathbf{D} = 1.5\hat{z}\ m/sec$ roughly translating to 6×10^6 particles. For rainfall simulation, we have used rainfall rate $50.0\,mm/hr$ with terminal velocity $2.0\,m/sec$ roughly giving 4×10^6 particles. For relevant calculations, we refer to [10]. The remaining parameters are same as those used by the respective papers.

Validation of Sensor Alignment

1) Weather Severity Alignment Verification: For ReDAM based fog alignment we obtain $\{\alpha^* = 0.01, \beta^* = 0.023 \times 1.6/MOR = 1.23 \times 10^{-4}\}$ for light fog, and $\{\alpha^* = 0.04, \beta^* = 0.023 \times 2.0/MOR = 6.13 \times 10^{-4}\}$ for dense fog. The parameters are evaluated over all samples in the corresponding split mentioned in STF dataset. The corresponding KS-Statistics values were obtained as $D_{KS} = \mathbf{0.092}$ and $D_{KS} = \mathbf{0.131}$ for light and dense fog, respectively, with the ECDF plots shown in Figure 3. Except at the jump locations, the ECDF shows a close agreement between the simulated fog with real fog in both light and dense conditions. For images, we use the Fréchet Inception Distance (FID) [16] & Kernel Inception Distance (KID) [17] metrics between simulated nuScenes foggy images and real STF images with the same fog level. We obtain the FID values of $\mathbf{4.2998}$ & $\mathbf{3.2772}$ and KID values of $\mathbf{0.1652}$ & $\mathbf{0.2091}$ for dense and light fog, respectively. The low value of these scores indicates that the generated images match closely with real images.



(a) Light fog, $D_{KS} = \mathbf{0.092}$ (b) Dense fog, $D_{KS} = \mathbf{0.131}$

Fig. 3. Comparison of ECDF plots of simulated and real weather. green and orange bands show intensity ECDF in real weather and red shows tuned simulated weather ECDF.

2) Particle Position Alignment Verification: To verify that weather particles occur at same location, we project these particles in LiDAR onto images and compute the error between projected position and corresponding rasterized position of image Gaussian. We find $\mathcal{N}_{comm}$ as the set of weather particles which remain visible (un-occluded) in both LiDAR and Camera. Then, we compute the re-projection error as

$$E = \sum_{i=1}^{|\mathcal{N}_{comm}|} \sum_{j=1}^{N_V} \|\mathbf{u}_{ij} - \hat{\mathbf{u}}(\mathbf{P}_j, \mathbf{X}_i)\|_2^2,$$

where N_V is the number of views of the camera, $\mathbf{X}_i$ is the ith lidar point, $\hat{\mathbf{u}}(\mathbf{P}_j, \mathbf{X}_i)$ is projection of $\mathbf{X}_i$ on jth view with projection matrix $\mathbf{P}_j$, and $\mathbf{u}_{ij}$ is the pixel coordinate of the rasterized location of the corresponding Gaussian. The low re-projection error values (< 1 px) shown in Table I validate the correctness of our position alignment.

TABLE I
RE-PROJECTION ERROR AND MMD DISTANCE

Weather	Fog	Snow	Rain
Re-projection error			
Mean error		0.7632	0.7641
Max error		1.3843	1.411
MMD Distance			
Aligned	**0.2837**	**0.3065**	0.1870
Unaligned	0.3491	0.3607	0.1870

3) Verification using fused features: To verify that aligned simulations are better than unaligned, we compare the MMD [18] distance of the fused LiDAR and Camera features between the real and simulated weather samples. We get Camera-LiDAR fused features by auxiliary weather classification model that uses concatenated LiDAR and Camera features and is trained to identify weather type. The lower value of MMD distance between the aligned and real shows a strong correspondence between them.

4) Verification using Perception task: We assess our alignment methods on the 3D detection task using three SOTA fusion models: BEVFusion (BEV-F) [19], Deep Interaction (DI) [20], and Cross-Modal Transformer (CMT) [21].

In Tables II and III, nuScenes-mini refers to the original clean version of data, nuScenes-mini-snow and nuScenes-mini-rain refers to the snow and rain simulated data on nuScenes-mini using Unified-weather-edit. nuScenes-mini-fog refers to fog simulation using ReDAM. Pretrained denotes the model trained on nuScenes clear-weather, Finetuned and Finetuned-mix refers to models finetuned on full simulated samples and a mix of 30% simulated and 70% clean samples.

We first compare the impact of alignment on 3-D detection tasks as shown in Table II. The unaligned versions consistently perform better than the aligned version by around $2-3\%$ mAP for rain and around 1% mAP for snow. This can be attributed to the fact that in the aligned dataset, at the positions corresponding to weather particles, information from both sensors is impacted. In contrast, in the unaligned version, partial information from at least one sensor is present.

Next, we show that our generated datasets can be used to achieve robustness to adverse weather. An ideal robust model would improve the performance on weather data (snow, rain, or fog) and retain the performance on the original clean data. As shown by the mAP and NDS scores in Table III, the pretrained BeV-Fusion performs very poorly on snowy and rainy data as compared to the clean data.

Next, we finetune the models on our aligned weather dataset and gain $\mathbf{0.13}, \mathbf{0.03}, \mathbf{0.07}$ mAP values, for snow, rain and fog, respectively. Finetuning the models only on simulated-weather datasets results in considerable drop in performance

TABLE II
PERFORMANCE OF SENSOR FUSION MODELS ON SNOW AND RAIN DATA GENERATED FROM UNIFIED-WEATHER-EDIT

nuScenes-mini-snow	Unaligned		Aligned	
	mAP	NDS	mAP	NDS
BEV-F	0.4034	0.4828	0.3943	0.4752
DI	0.4139	0.4626	0.4010	0.4541
CMT	0.4007	0.4800	0.3965	0.4760
nuScenes-mini-rain	Unaligned		Aligned	
	mAP	NDS	mAP	NDS
BEV-F	0.4933	0.4384	0.4649	0.5192
DI	0.5206	0.5294	0.4926	0.5133
CMT	0.4958	0.5316	0.4766	0.5207

in clear datasets. Therefore, we finetune our models on a mix of weather data and clean data. It is evident from the results that the mix finetuning performs the best.

TABLE III
MAP AND NDS OF BEV FUSION ON SNOW, RAIN AND FOG SIMULATIONS

Snow	nuScenes-mini		nuScenes-mini-snow	
	MAP	NDS	MAP	NDS
Pretrained	0.5758	0.5835	0.3943	0.4752
Finetuned-snow	0.5394	0.5618	**0.5161** ↑	0.5471
Finetuned-snow-mix	0.5606	0.5756	0.5103	**0.5472** ↑
Rain	nuScenes-mini		nuScenes-mini-rain	
	MAP	NDS	MAP	NDS
Pretrained	0.5758	0.5835	0.4646	0.5192
Finetuned-rain	0.5468	0.5685	**0.4909** ↑	**0.5287** ↑
Finetuned-rain-mix	0.5590	0.5765	0.4710	0.5201
Fog	nuScenes-mini		nuScenes-mini-fog	
	MAP	NDS	MAP	NDS
Pretrained	0.5758	0.5835	0.4314	0.4991
Finetuned-fog	0.4448	0.5129	**0.5018** ↑	**0.5407** ↑
Finetuned-fog-mix	0.5515	0.5725	0.4861	0.5301

IV. CONCLUSION AND FUTURE WORK

We propose ReDAM and Unified-weather-edit to simulate aligned Camera and LiDAR simulations under adverse weather. These alignment methods output more realistic simulations, which can be used by fusion models to achieve robustness via finetuning on the simulations. Unlike the overly optimistic results observed with unaligned data, our simulations present a more grounded assessment.

ReDAM and Unified-weather-edit are capable to generate aligned simulations at different weather severity levels. In the future, we plan to provide extensive simulations at different levels for larger dataset (nuScenes trainval). Further, the proposed alignment strategy is easily generalizable to other datasets, sensing modalities (like radar) and other faults (like mixed weather scenarios) which we intend to integrate in future studies.

REFERENCES

[1] C. Qian, W. Li, Y. Guo, and G. Markkula, "Weatheredit: Controllable weather editing with 4d gaussian field," 2025. [Online]. Available: https://arxiv.org/abs/2505.20471

[2] Y. Huang, K. Yu, Q. Guo, F. Juefei-Xu, X. Jia, T. Li, G. Pu, and Y. Liu, "Improving robustness of lidar-camera fusion model against weather corruption from fusion strategy perspective," *arXiv preprint arXiv:2402.02738*, 2024.

[3] Y. Dong, C. Kang, J. Zhang, Z. Zhu, Y. Wang, X. Yang, H. Su, X. Wei, and J. Zhu, "Benchmarking robustness of 3d object detection to common corruptions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1022–1032.

[4] X. Hao, G. Liu, Y. Zhao, Y. Ji, M. Wei, H. Zhao, L. Kong, R. Yin, and Y. Liu, "Msc-bench: Benchmarking and analyzing multi-sensor corruption for driving perception," *arXiv preprint arXiv:2501.01037*, 2025.

[5] M. Hahner, C. Sakaridis, M. Bijelic, F. Heide, F. Yu, D. Dai, and L. Van Gool, "Lidar snowfall simulation for robust 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 364–16 374.

[6] M. Hahner, C. Sakaridis, D. Dai, and L. 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*, 2021, pp. 15 283–15 292.

[7] V. W. Berger and Y. Zhou, *Kolmogorov–Smirnov Test: Overview*. John Wiley Sons, Ltd, 2014. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat06558

[8] R. H. Rasshofer, M. Spies, and H. Spies, "Influences of weather phenomena on automotive laser radar systems," *Advances in Radio Science*, vol. 9, pp. 49–60, Jul. 2011.

[9] M. Hahner, C. Sakaridis, D. Dai, and L. V. Gool, "Fog simulation on real lidar point clouds for 3d object detection in adverse weather," 2021. [Online]. Available: https://arxiv.org/abs/2108.05249

[10] M. Hahner, C. Sakaridis, M. Bijelic, F. Heide, F. Yu, D. Dai, and L. V. Gool, "Lidar snowfall simulation for robust 3d object detection," 2022. [Online]. Available: https://arxiv.org/abs/2203.15118

[11] V. Kilic, D. Hegde, V. Sindagi, A. B. Cooper, M. A. Foster, and V. M. Patel, "Lidar light scattering augmentation (lisa): Physics-based simulation of adverse weather conditions for 3d object detection," 2021. [Online]. Available: https://arxiv.org/abs/2107.07004

[12] L. Jiang, C. Zhang, M. Huang, C. Liu, J. Shi, and C. C. Loy, "Tsit: A simple and versatile framework for image-to-image translation," in *European conference on computer vision*. Springer, 2020, pp. 206–222.

[13] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide, "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[14] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," 2020. [Online]. Available: https://arxiv.org/abs/1903.11027

[15] C. Weitkamp, Ed., *Lidar: Range-Resolved Optical Remote Sensing of the Atmosphere*, 1st ed., ser. Springer Series in Optical Sciences. New York, NY: Springer, 2005.

[16] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," 2018. [Online]. Available: https://arxiv.org/abs/1706.08500

[17] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," 2021. [Online]. Available: https://arxiv.org/abs/1801.01401

[18] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The journal of machine learning research*, vol. 13, no. 1, pp. 723–773, 2012.

[19] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, and Z. Tang, "Bevfusion: A simple and robust lidar-camera fusion framework," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10 421–10 434, 2022.

[20] Z. Yang, J. Chen, Z. Miao, W. Li, X. Zhu, and L. Zhang, "Deepinteraction: 3d object detection via modality interaction," *Advances in Neural Information Processing Systems*, vol. 35, pp. 1992–2005, 2022.

[21] J. Yan, Y. Liu, J. Sun, F. Jia, S. Li, T. Wang, and X. Zhang, "Cross modal transformer: Towards fast and robust 3d object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 18 268–18 278.