ROADVIEW

Robust Automated Driving in Extreme Weather

Project No. 101069576

Deliverable 3.3

Library of validated statistical noise models

WP3 – Validated sensor and sensor noise models for synthetic environments

Authors	Pierre Duthon,Hamed Ouattara, Sébastien Liandrat (CE), Yuri Poledna (THI)
Lead participant	CE
Delivery date	29 May 2024
Dissemination level	Public
Type	Report

Version 01

Co-funded by the European Union

Innovate UK

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them. Project grant no. 101069576.

UK participants in this project are co-funded by Innovate UK under contract no.10045139.
Swiss participants in this project are co-funded by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract no. 22.00123.

Revision history

Author(s)	Description	Date
Pierre Duthon (CE)	Draft deliverable	03/05/2024
Yuri Poledna (THI)	Revision 1	06/05/2024
Pierre Duthon (CE)	Revision 2	14/05/2024
Martin Sanfridson (RISE)	Revision 3	21/05/2024
Pierre Duthon (CE), Eren Aksoy (HH)	Final version	24/05/2024
Andreia Cruz (accelCH)	Formal check	29/05/2024

List of Figures

List of Tables

Partner short names

LUA	Lapin Ammattikorkeakoulu OY
THI	Technische Hochschule Ingolstadt
CE	Centre d'études et d'expertise sur les risques, l'environnement, la mobilité et l'aménagement
WMG	The University of Warwick
RISE	RISE Research Institutes of Sweden AB

Abbreviations

BEV	Bird Eye View
CUT	Contrastive Unpaired Images
D	Deliverable
GAN	Generative Adversarial Networks
LiDAR	Light Detection And Ranging
MOR	Meteorological Optical Range
PDD	Puy-de-Dôme
RADAR	RAdio Detection And Ranging
RCS	RADAR Cross section
T	Task
WP	Work Package

Executive summary

Objectives

The objective of this deliverable is to describe and justify the data-driven weather noise model library published as part of D3.3. This library is published on the project's official GitHub (available at https://github.com/roadview-project/data-driven-noise-models). It covers fog, rain, and snow conditions and Light Detection And Ranging (LiDAR), camera and RAdio Detection And Ranging (RADAR) sensors.

Methodology and implementation

The data-driven noise models are based on the following principle: natural data with and without degraded weather is used to determine the noise added by the weather using a machine learning method. The decision was made to treat the case of camera, LiDAR and RADAR independently, as the input data is very different. For cameras, a Cycle-GAN method was used to process the images in their entirety. For LiDAR and RADAR, a point-by-point classification method based on the notion of vanishing distance was created. The training data used are public images collected by CE in the case of the cameras, and data acquired by CE on the Puy-de-Dôme (PDD) site (France) during the 2022-2023 and 2023-2024 winters (created as part of T3.2).

Outcomes

Concerning cameras, the Cycle-GAN method was implemented for the different weather conditions, with the possibility of giving intensity classes. Some of the results show the relevance of data-driven models, which in the case of snow make it possible to simulate all the associated effects. This is a counter point to just simulating the snow falling from the sky. For example, this method allows the removal of leaves from trees, snow on the ground including vehicle tracks. Concerning LiDARs, the point-by-point method based on the vanishing distance makes it possible to reproduce the three weather conditions, by specifying the exact desired intensity of the weather. Finally, this report shows that RADAR is insensitive to weather for the tested adverse weather intensities.

Next steps

The next step is to validate the models developed during T3.4. This validation will be carried out on data from the REHEARSE database (created as part of T3.2). This work will be promoted through the publication of scientific articles on the various models developed. Finally, some data and camera models prepared in Task 3.3 will be reused in Task 5.2 on camera-based measurement of weather conditions.

1 Introduction

The perception system used in the ROADVIEW project consists of cameras, LiDAR, and RADAR. WP3 aims to provide a complete simulation chain, spanning from perception to vehicle dynamics on one hand, and including adverse weather conditions on the other. WP3 aims to propose two variants of noise models: a physics-based variant (T3.4) and a data-driven variant (T3.3). These models adverse weather conditions considered are fog, rain, and snow. In this, we propose here three noise models (camera, LiDAR, and RADAR) that consider these three weather conditions.

The physic-based models use physical laws to simulate the impact of weather conditions. This document then focuses solely on the data-driven solution. The data-driven models are based on optimization methods (machine learning) using real data with and without degraded weather conditions. From a "clear weather" sub-dataset and a "degraded weather" sub-dataset (i.e., fog, rain, or snow), it is indeed possible to learn the impact of harsh weather on the sensor. As shown in Figure 1, the principle of the data-driven model is the same regardless of the sensor.

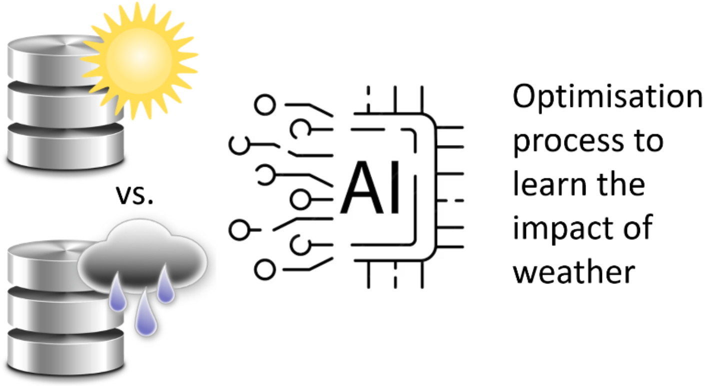


Figure 1: Schematic diagram of data-driven models. By comparing data with different weather conditions, the algorithm can learn to simulate the weather.

The modelled sensors (camera, LiDAR, and RADAR) have different sensing principles, therefore, different models have been chosen for each. Cameras, on one hand, use pixel matrices, containing colour information and no distance information, with very high spatial coherence. Conversely, LiDAR and RADAR use point clouds, unordered, containing only a reflectivity value (in the case of LiDAR, Radar has RADAR Cross section (RCS) and velocity) and distance information, with relative spatial coherence. Moreover, in the case of cameras, it is possible to access large image databases (as will be described later), whereas for LiDAR and RADAR (specially the 4D RADARs used by the ROADVIEW project), there are very few public datasets available, and we must therefore optimize models from our own database, limited in volume.

Furthermore, LiDAR and RADAR, not being based on the same wavelengths, have completely different characteristics in terms of precision/sensitivity to weather. Thus, the decision was made to create independent and fundamentally different models for the three modalities.

This report presents the data-driven noise models developed as part of the ROADVIEW project. Particularly, it describes the library published as part of D3.3. This library is published on the project's official GitHub[1], deployed as part of the project's WP3. After training, the noise models obtained are compared visually with real data. This comparison was carried out both on the databases collected by CE and on the REHEARSE database (created as part of task 3.2 - D3.2). The precise validation of the models, using dedicated metrics, will then be carried out as part of task 3.4 with data from the REHEARSE database. This report is divided into three parts, one for each sensor model created. Each part is equally divided into 6 subparts, starting with a literature review, followed by the model description, and analysis of available datasets, the results follow the discussion of the implementation and finally the discussion on the model.

[1] https://github.com/roadview-project/data-driven-noise-models

2 Camera

In this part, the goal is to realistically change the weather of a clear weather camera image by adding rain, snow, or fog at various intensities. It is important here to specify what we mean by realistic. As mentioned in the introduction, there are two main methods for altering the weather in an image: physic-based methods and data-driven methods. Both approaches model the weather phenomenon and then use computer vision techniques to replicate it on an image.

Physic-based methods examine how the weather phenomenon affects light reflected by objects, based on specific physical laws, and then propose a model that can be coupled with computer vision techniques to reproduce the phenomenon on images. This approach starts with a certain understanding of the underlying phenomenon causes. In this sense, it contrasts with the second family: data-driven methods.

Data Driven methods completely abstract the causes and adopt a machine learning approach, treating the weather phenomenon as a random experiment whose universe is the set of all possible manifestations or realizations of the phenomenon.

In this work, since we observe these manifestations through images captured by cameras, the universe includes all possible images of the weather phenomenon, such as all possible images of rain. Statistical modelling then aims to find the law of this random experiment, i.e., a function to estimate the probability of producing a given rain image. In theory, this involves using a sample of rain images and statistical functions to infer this law. These estimators must be consistent, meaning that the larger the sample size, the closer we get to the true law of the data. Once this modelling is achieved, computer vision methods can be used to reproduce the weather phenomenon. This is the approach we focus on. The weather phenomenon reproduction on an image is realistic if it is close to what we would likely have obtained if the image had been captured by a camera at the moment the phenomenon occurred.

We will show that data-driven modelling poses certain challenges, but a number of paradigms based on deep neural network learning enable us to reproduce weather phenomena while avoiding us from statistically evaluating the realistic nature of the results obtained. The following section proposes a literature review, followed by the model description, and analysis of available datasets. Section 2.5 presents the results before the discussion on the model.

2.1 Literature Review

In general, in a scene where it rains, objects tend to be much more reflective because they are soaked with water. Similarly, in a scene where it snows, they will be covered with snow (white appearance), etc. Just as two painters can represent the exact same scene with styles unique to each (appearance of objects), two different weather conditions give the same scene different appearances or styles. Thus, our issue fits into a well-known method in artificial intelligence and computer vision, known as style transfer.

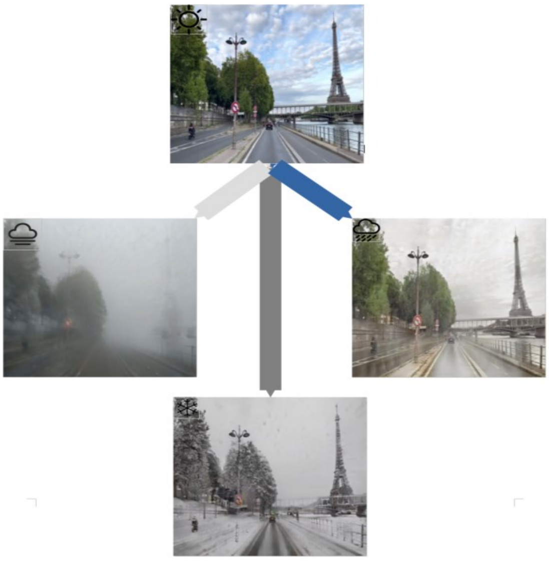


Figure 2: The aim of the data-driven camera model is to switch from clear weather to different harsh weather conditions (fog, rain and snow). Sun image is real, fog, rain and snow images are generated.

Figure 2 illustrates the aim of data-driven noise model for camera. In this figure, we can see the exact same scene with only the weather condition differing. The observation is clear: the objects or things present in the three scenes remain the same, what changes is their appearance.

Style Transfers involves moving from an image expressed in one style to another while preserving the content. This implies, in each image, being able to distinguish what pertains to the content from what pertains to the style or appearance of the objects present in the image. Recently, this field has seen rapid advancements, as shown in Table 1. In 2016, the method proposed by Gatys et al. [1] allows changing the style of an image to the style of a reference image using the intermediate layers of a neural network.

Table 1: Summary of transfer style methods identified in the literature.

Algorithm name	Year	REF	Paired database	Weather-specific	Type
Neural style transfer	2016	[1]	Yes	No	GAN
Pix2Pix	2017	[2]	Yes	No	GAN
CycleGAN	2017	[3]	No	No	GAN
CUT	2020	[4]	No	No	GAN
Weather GAN	2021	[5]	No	Yes	GAN
TPSeNCE	2023	[6]	No	Yes	GAN
CYCLE-DIFFUSION	2022	[7]	No	No	DIFFUSION
Zero-shot contrastive loss	2023	[8]	No	No	DIFFUSION
GInStyle	2023	[9]	No	Yes	DIFFUSION

In a neural network, the first layers can extract features related to the style of the image while those closer to the output capture features related to the objects. Thus, by passing two images of different styles through the same neural network, we can dissociate their style from their content and then make a mix by linking the style of one to the content of the other. This method, named "neural style transfer," produces visually interesting results but is not suitable for the task we aim at. Indeed, as we have already mentioned, a good statistical/machine learning method is one that improves as the quantity of data increases. However, with "neural style transfer," increasing the data quantity will have no effect, as this method can only transfer the style of one image to another.

Other algorithms have been developed to allow the style transfer of a group of images to another group. The neural network Pix2Pix [2] is one of the first successful methods of style transfer using Generative Adversarial Networks (GAN). An example of this is to transform sunny images into rainy ones. For this operation, we start by creating a database containing the same scenes photographed both in good weather and under rain, as shown in Figure 3.

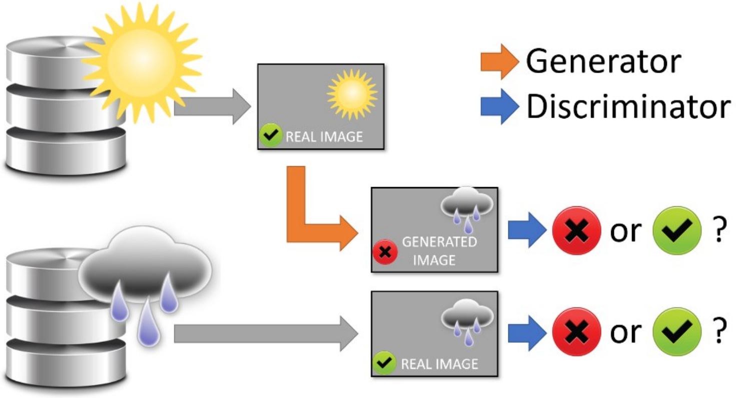


Figure 3: How a GAN works. Using two databases with different styles, a generator creates generated images, while a discriminator determines whether the image is real or fake.

In this system, the GAN discriminator analyses pairs of real images — one sunny and the other rainy. It learns not only to recognize that rainy images are realistic but also to ensure that both images in each pair represent the same scene. At this point, the generator comes into play: it takes the sunny image and transforms it into a rainy version. This transformation is guided by the teachings of the discriminator, which checks that the transformed image remains true to reality and the original scene, thus ensuring content coherence.

Pix2Pix is considered very effective among style transfer methods based on GANs. However, the major difficulty lies in creating the necessary databases. For example, for a style transfer from horse to zebra, you must find pairs of images of horses and zebras that resemble each other except for the colour of their fur. Similarly, recreating exactly the same scene with only weather variations can be complex. The algorithm needs to be able to identify variations linked solely to the weather. Conversely, the algorithm must not modify the other stylistic elements of the image, such as buildings, vegetation, etc.

We have thus explored algorithms capable of transferring the style of one group of images to another without requiring corresponding pairs of images. For example, these algorithms can transform sunny images taken in Paris using rainy images from Rome. Two techniques exist: the "contrastive-unpaired-images" (CUT) [4] and the CycleGAN [3]. The CUT works using both GAN and contrastive learning. This approach uses the conditioning of the GAN to change the style of images while preserving their content through contrastive learning. Whilst in the domain of style transfer, the contrastive aspect is implemented by comparisons made between selected pieces of a source style image and those transformed by a GAN. This process aims to make the content of the transformed pieces similar to the originals, while ensuring they are distinctly different from those from other images in the database.

CycleGAN uses two distinct GANs, for example: one transforms sunny images into rainy ones, and the other does the reverse. This approach allows for an efficient transfer between two styles. To preserve the original content of the images during the transfer, CycleGAN incorporates a technique called 'cyclic coherence error'. This technique is based on the idea that applying the two transformations (sunny to rainy, then rainy to sunny), the resulting image should resemble the original image. We have experimented with these methods to change the weather aspect of images.

Our tests reveal that although both algorithms allow style transfer, they can also create artifacts in the images. However, images from CycleGAN method are of better quality than those produced by CUT method. Zheng et al. [10] present an improved version of the CUT that minimizes artifacts and improves rain intensity control, but this technique requires image segmentation, making the process complex. Similarly, Li [5] proposes a method based on GANs that aims to change the weather in an image using attention mechanisms. These methods, focusing on the same issue as ours regarding weather simulation, are interested in extracting weather features in the downstream image. These two recent methods focus more on working out the structure of the networks used than on the learning process. On the other hand, in order to take better advantage of the training data, and to get as close as possible to a paired learning database, like in Pix2Pix, we have developed an innovative learning method called 'Paired Image Subgroups to Image Subgroups' (Group2Group). This method, which optimizes data selection for learning, significantly reduces artifacts, both for CycleGAN and CUT models. Moreover, it opens the door to various applications in style transfer, such as transferring a style to several merged styles or step-by-step denoising.

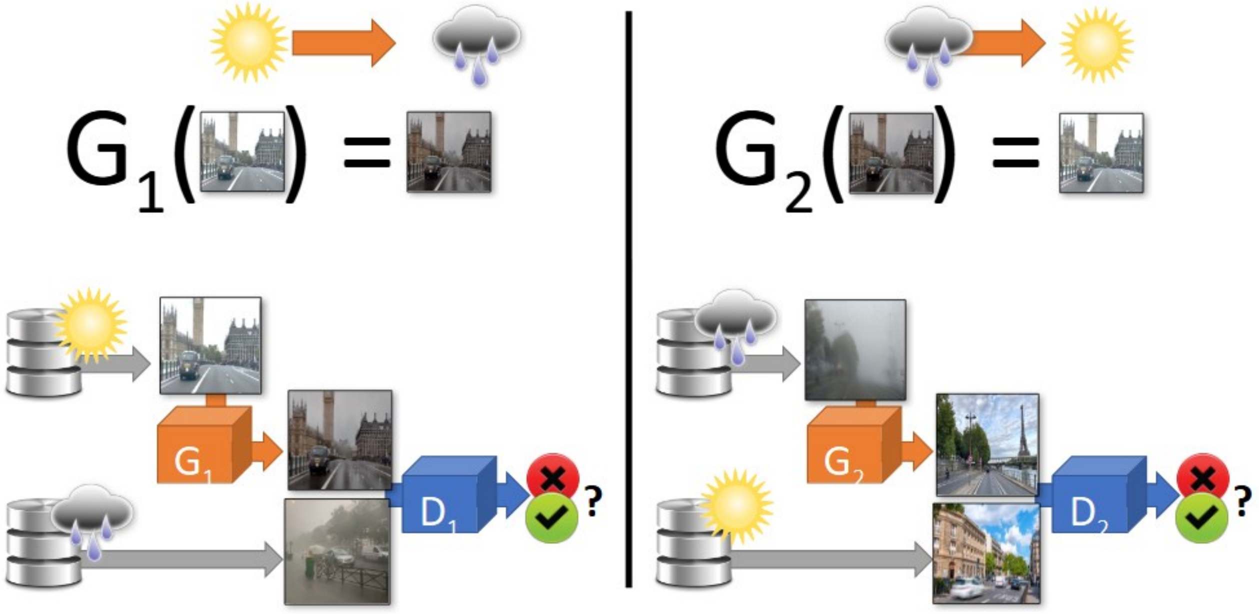
There is another category of transfer-style methods: diffusion models. Yang et al. [8] propose an approach that combines Zero-Shot learning and contrastive learning to perform style transfer using diffusion algorithms. This second category of models seems promising and offers beautiful perspectives. However, these methods are generally more cumbersome than methods based on GANs and we will see that the latter are sufficient to solve the problem posed. In addition, the use of the GAN discriminator will enable us to detect weather conditions by camera, as part of task 5.2. This explains the decision to leave aside diffusion models and to focus on GAN methods here.

In conclusion, our literature review has led us to adopt the CycleGAN model with the method Group2Group to develop our models aimed at modifying the weather on an image. These two elements are described in detail in the next section.

2.2 Model Structure

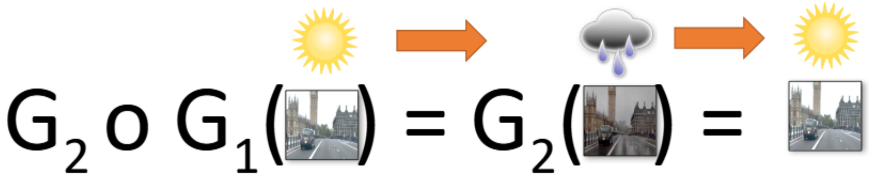
2.2.1 CycleGAN Model

The CycleGAN model proposed by Zhu et al. [3] enables the transition between sets of images of different styles, such as a set of rainy images and another of sunny images. It utilizes two GANs to convert images from one set to the other. For example, one GAN transforms sunny images into rainy images, and the other does the reverse, as shown on Figure 4. To preserve the content of the images while changing the style, the model incorporates a "cyclic consistency error", as shown on Figure 5. This error ensures that the composition of the mapping functions, performed by the GAN generators, is identical to the identity.



$$L_{GAN}(G,D) = \mathbb{E}_{p_{data}}(\log D(x)) + \mathbb{E}_{p_z}[\log(1 - D(G(z))]$$

Figure 4: CycleGAN method operating principle



$$L_{cycl}(G_2, D, X, Y) = \mathbb{E}_{x\sim P_x}(\|G_2(G_1(x)) - x\|_1) + \mathbb{E}_{y\sim p_y}(\|G_1(G_2(y)) - y\|_2)\,.$$

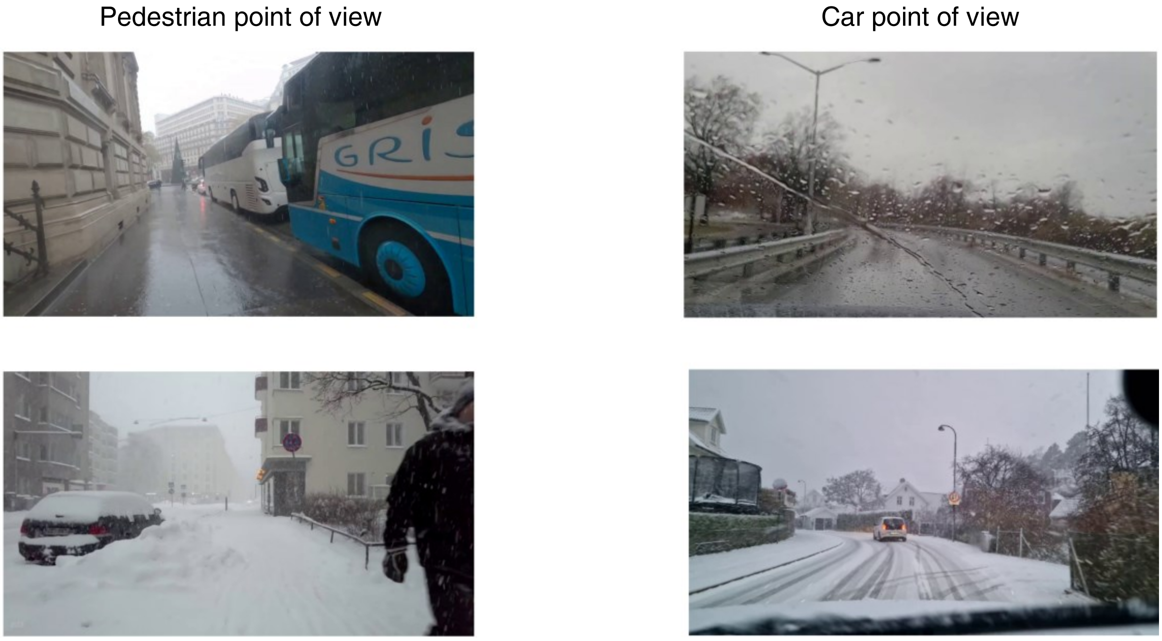
Figure 5: CycleGAN method operating principle: cyclic consistency error

The structure of the CycleGAN that we used for our training is the one described in [3], and the learning error used is “Least Squares GAN” [11].

One can say that the goal of the CycleGAN is to find an isomorphism between the two data sets. However, in our case, by taking two images at random from each of the two groups in the database (one in clear weather, and the other in rainy conditions, for example), it happens that the elements that differ are not only linked to the weather. For example, this may be the case when one image contains trees and the other does not. This can bias the learning, as discussed in the section 2.4.2. Then, we propose a method of selecting training data that reduces possible isomorphisms and as we will show, significantly improves the simulation results. This method is presented in the next section.

2.2.2 Paired Image Subgroups to Image Subgroups (Group2Group) method

As previously mentioned, when wishing to apply a style transfer between two data sets, two main methods using GANs are available: one using paired images with the Pix2Pix network [2] and the other using unpaired images with CUT [4] or CycleGAN [3]. The unpaired methods simply require two groups of images of different styles. However, for the ROADVIEW project, the weather conditions we wish to simulate must be specific to road contexts and viewed from a vehicle. This significantly limits the usable data among those we have collected. Since data is crucial in artificial intelligence, exploiting all available data is a major advantage, especially in a project aiming to simulate reality. We have therefore chosen to use all available data, whether road-related or not, to train an initial model. Then, we will refine this model by specializing it with data only from road scenes and from the perspective of a vehicle. This approach, common in artificial intelligence, tends to yield better results especially when data are limited. However, we encountered a major problem: the perception of weather conditions like rain or snow differs significantly between pedestrians and vehicles, as illustrated in Figure 6.



Pedestrian point of view

Car point of view

Figure 6: Difference in point of view and impact of the weather for an image from the point of view of a pedestrian and a vehicle.

Training methods for CUT and CycleGAN networks can introduce mixes between different viewpoints and contexts during style transfer, as shown in Figure 7 (forest vs. city). We believe this can cause artifacts and sometimes distortions. Indeed, if during a training period, the network learns to switch from sunny weather conditions to rainy, while changing from a pedestrian perspective to that of a vehicle, these viewpoint changes can affect the final style, particularly creating distortion effects, as shown on Figure 8. In this figure, for example: buildings appear in the sky, red reflections appear all over the floor.

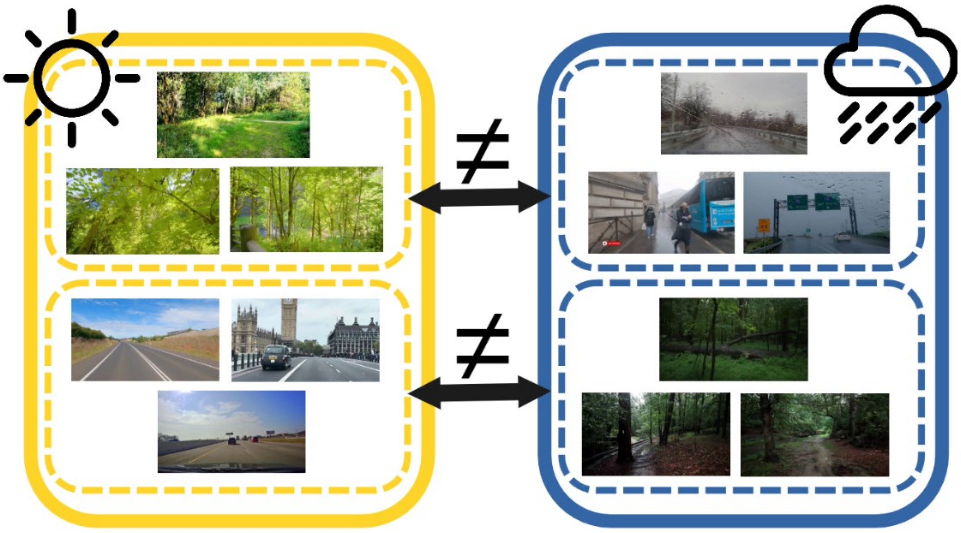


Figure 7: Example of a group of images that pose a problem during transferring styles with images from different sources.

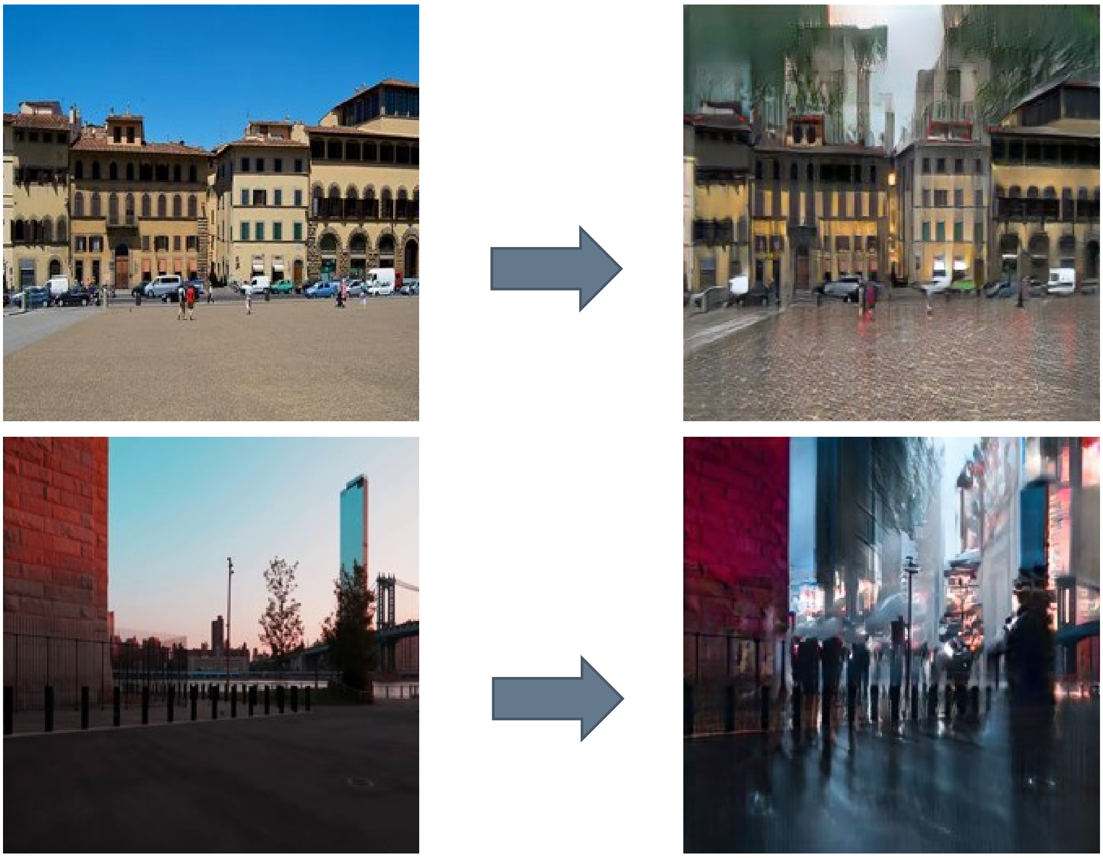


Figure 8: Example of distortions obtained with training in the Group2Group method.

The concept of style is quite broad. It encompasses aspects such as viewpoint, the structure of buildings, etc. A particularity of style transfer algorithms, like CUT or CycleGAN, is their ability to automatically determine the elements that characterize style and those that define content in two data sets. Generally, if a subset of data includes images that share common features — not related to the weather, such as the angle of the shot or the architecture of buildings, or even the dominant presence of greenery in a forest — these elements are considered as part of the style of that subset. For example, whether the weather is sunny or rainy, buildings in Paris typically have a

Haussmann style, while in London, they are mostly Victorian. If our algorithm associates these characteristics during learning, it could learn to recognize and transfer these architectural styles.

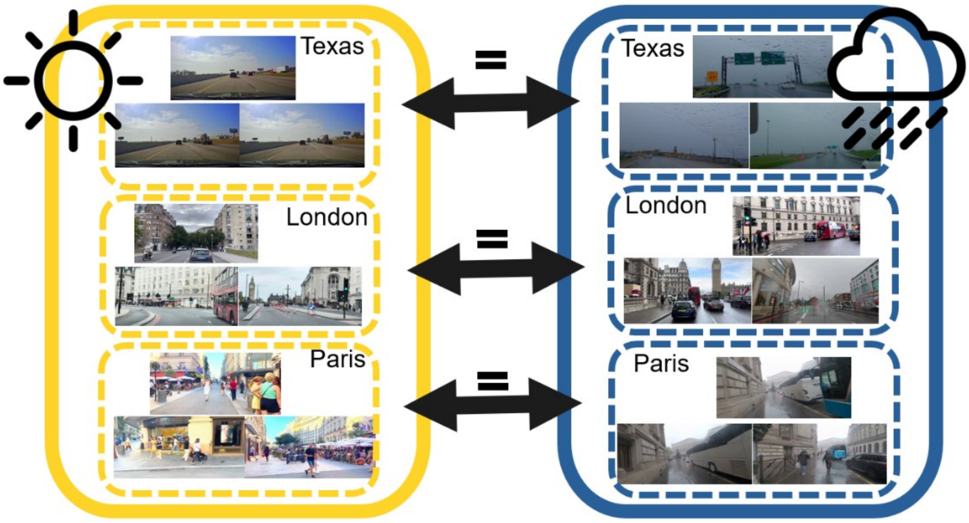


Figure 9: Parallelization of subgroups with specific places captured in different weather conditions

Our tests have revealed that, in certain scenarios, the style transfer of an image is affected by the specific subgroups to which it belongs and by the associations that form fortuitously during training. It is crucial to note that ideally, the only differences between the subgroups encountered during training should concern the weather conditions. This condition is essential because algorithms like CUT or CycleGAN, which automatically determine the constitutive elements of style, are not ideally suited for tasks where the style to be transferred is as specific as weather conditions. During training, we thus create precise correspondences between different subgroups arising from two distinct weather conditions. Ideally, these subgroups must be differentiated solely by their weather conditions. We have named this method "Paired Image Subgroups to Image Subgroups" (Group2Group), as shown in the Figure 9.

This method of selecting pairs for training ensures that the main difference between two subgroups is their weather condition. We will demonstrate further that learning based on these subgroups forces the network to condition the style transfer to the subgroups of origin and destination. This approach not only improves the realism of the results obtained but also opens the door to new possibilities previously unexplored: such as transferring two styles to a group of images, for example assigning rain and snow to a group of images, enhancing denoising by training through denoising with multiple styles, ...

2.3 Dataset

For training the camera data-driven models, we utilize two main types of databases: public databases from research similar to ours, and those developed by CE. Among the public databases, we particularly use Image2weather [12], which contains 183,798 images, and the Multi-Class Weather Dataset [13], with 1,094 images. However, the disorganized format of these databases presents a challenge for our Group2Group learning method, as it is difficult to form subgroups differentiated primarily by their weather conditions. To address this issue, we have created two new databases tailored to our method, both covering four weather conditions. The first one containing approximately 700,000 images from roadside cameras. The second one is created from videos sourced from the internet, from vehicle and pedestrian point of views. The second database we created contains about 1,145,995 images. Figure 10 presents examples of images from this vehicle and pedestrian cameras database. Thanks to these resources, we can form image groups that show the same location under different weather conditions, as illustrated in Figure 11.

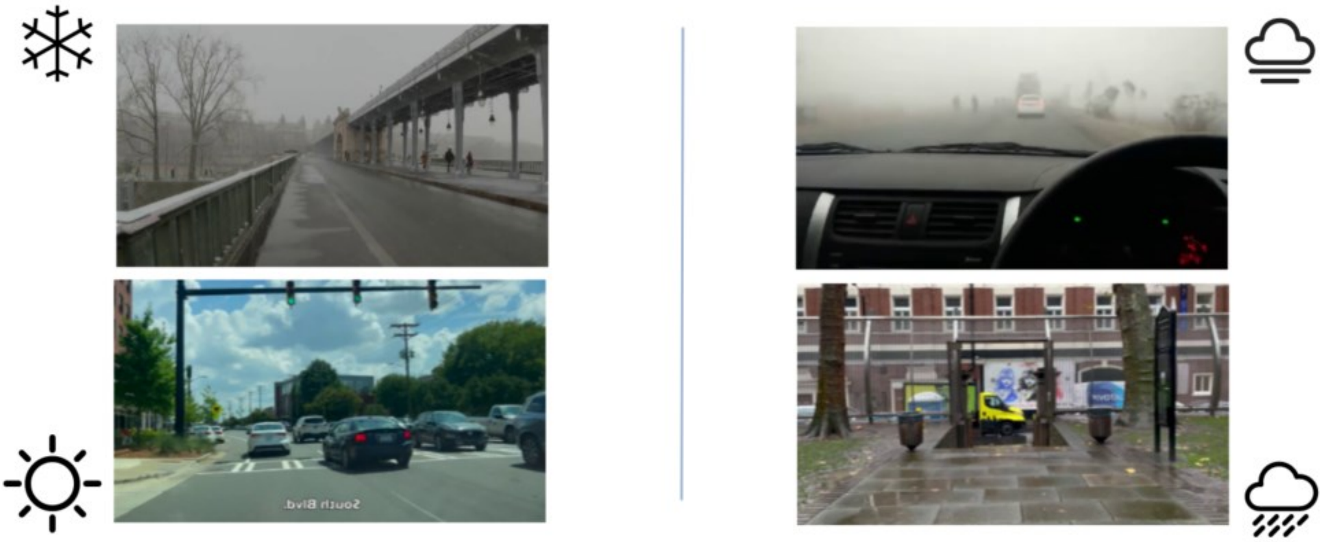


Figure 10: Some images of CE's vehicle and pedestrian database

Another part of the task was to control the intensity of the weather conditions. To do this, we had to annotate our images to distinguish different weather intensities. Table 2 summarizes the databases used to train the data-driven camera noise models. The weather intensity classes have been defined by a panel of experienced observers. They are as follows. For clear: clear sky corresponding to blue sky without any cloud, overcast corresponding to grey sky or blue sky with some clouds. For rain: wet ground only (rainfall rate 5mm/h), medium (rainfall rate 20mm/h), and heavy (rainfall rate 40mm/h). For fog: light (Meteorological Optical Range (MOR) 400m), medium (MOR 200m), and heavy (MOR 30m). For snow: ground covered by snow only (rainfall rate 0mm/h), and heavy (rainfall rate 10mm/h). The validation database, which is completely independent, is the REHEARSE database. This database is not presented here because it was created as part of task 3.2 and will be used to validate the models as part of task 3.4.

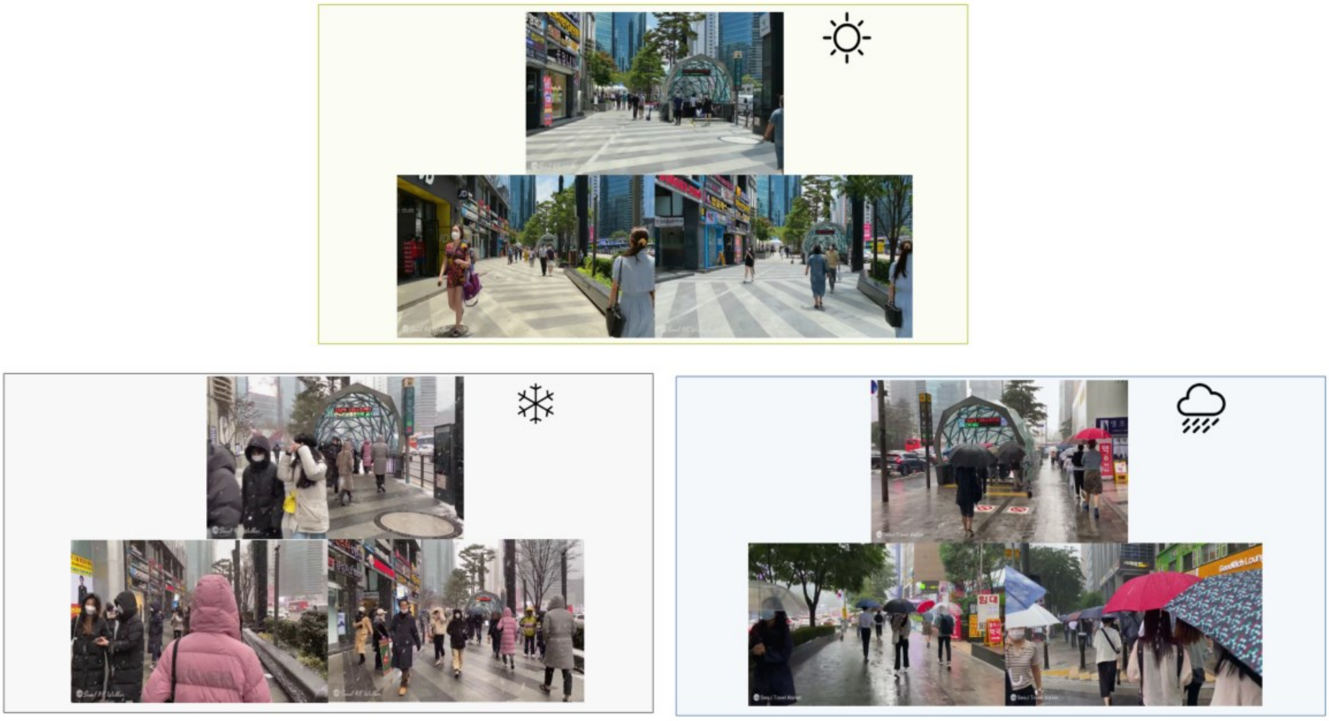


Figure 11: Group of images showing the same place in different weather conditions

For each training session, we have created a database with different subgroups. In these subgroups, we try to parallelize the scenes as much as possible. For example, sunny images of Paris vs. rainy images of Paris. Sunny forest image in Germany vs. snowy forest image in Germany.

Database	Clear		Fog			Rain			Snow	
	Clear sky	Overcast	Light	Medium	Heavy	Wet ground	Medium	Heavy	On floor	Heavy
MCWD [12]	500		500			500			500	
Image2Weather [13]	70501	45662	357			1369			1252	
Pedestrian	43035	208285	23818	23713	13840	5302	77740	72787	4764	26358
Vehicle	82820	224239	6147	14390	22249	3242	26075	54175	2561	12744
Roadside	~ 250000	~ 250000	~ 250000			~ 250000			~ 250000	

Table 2: Summary of image databases used to train data-driven camera noise models.

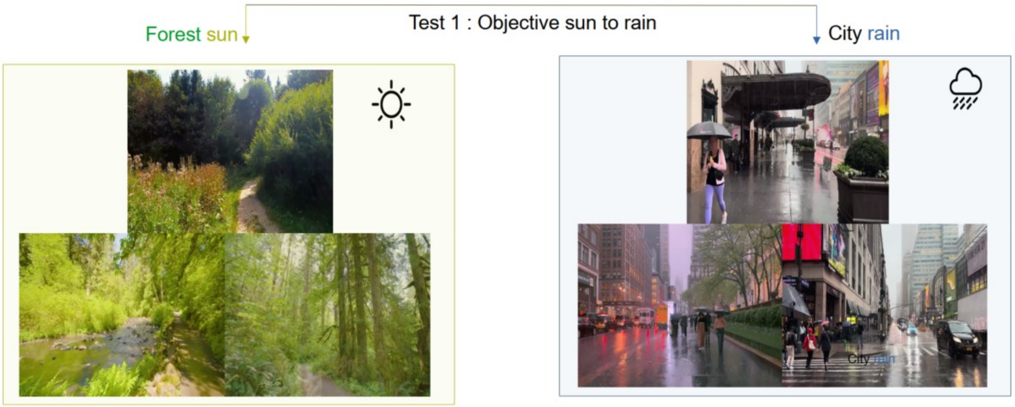
2.4 Implementation details

2.4.1 Technical details

All the details about the model as well as the amount of data used and learning parameters are given on the GitHub repository. For some of the models, we first trained on all our data from pedestrian and vehicle viewpoints, and then, we fine-tuned the model on vehicle-type data only. In general terms, the models have 256x256 pixels input and output images.

2.4.2 Sub-group method benefit

The goal of this test is to demonstrate that algorithms struggle with style transfer when the contexts are very different. Thus, we created three datasets, each containing 700 items: the first groups images of a forest in clear weather (no precipitation), the second images of a large city in the rain, and the third images of the same city in clear weather (Figure 12).



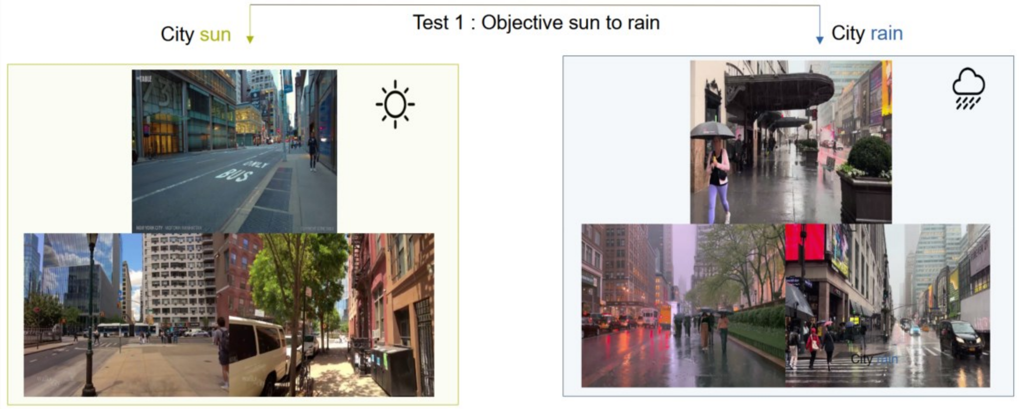


Figure 12: Data used to demonstrate the need for the Group2Group learning method.

When trying to transfer from the forest (sun) to the city (rain), we get the results in Figure 13 with the CycleGAN algorithm (CUT gives a similar result) then from the city (sun) to the city (rain). There are many observations, the results are generally of poor quality when transferring from the forest to the city and vice versa. It is observed that from the forest to the city the style transfer adds a lot of green. On the contrary, from city to city the results are generally of better quality.

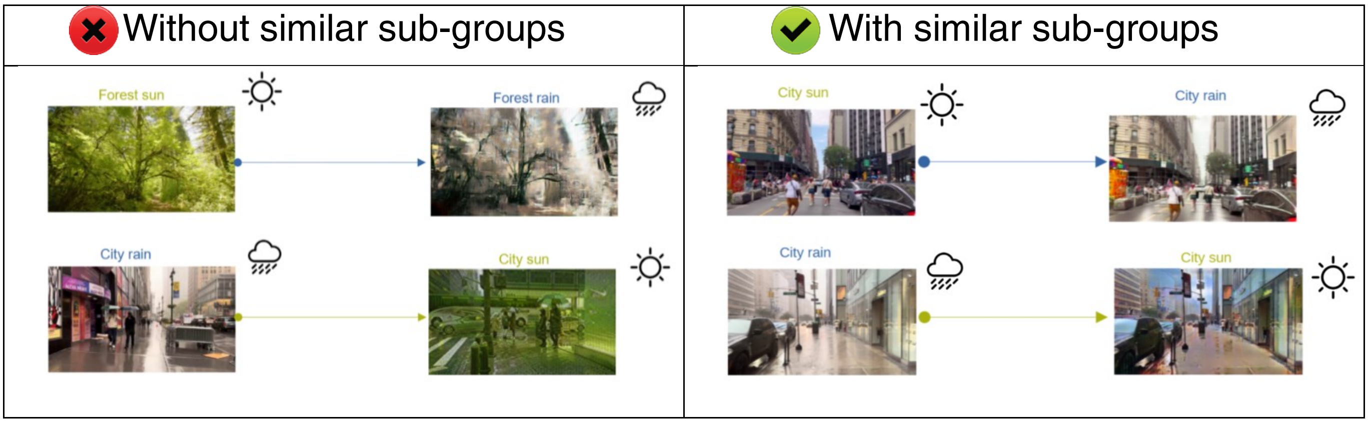


Figure 13: Results obtained with group learning method on forest vs city data.

Then, to achieve a successful style transfer, it is not enough to simply have sunny images on one side and rainy images on the other. The quality of the transfer decreases when the two groups of images are very different. As shown in Figure 13, when trying to transform rainy city images into sunny forest images, the excessive presence of green reveals that the algorithm associates this colour with the stylistic aspect of forests. This experiment also shows that the more similar the groups of images are, the more effective the transfer process is. However, with techniques like CycleGAN or CUT, there is a risk that two very different subgroups are randomly associated during training. Ideally, in our study, the image groups should differ only by the weather conditions, in line with the goal of the Group2Group method.

2.5 Results

We present the results obtained with the Group2Group method for three types of weather conditions: rain, snow, and fog. For rain, we analysed three different intensities; for snow, two intensities; and for fog, three intensities.

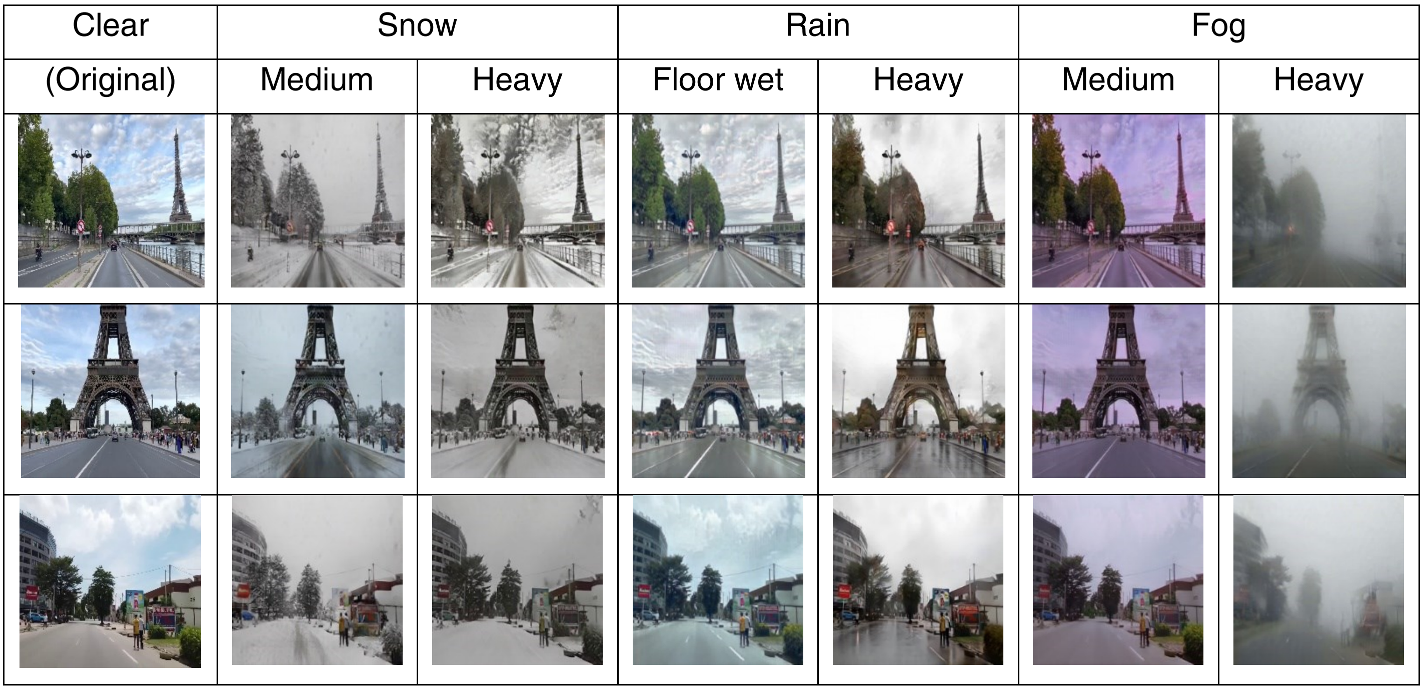


Figure 14: Results obtained on some images from the CE vehicle database

Upon examining the results in more detail, we notice many interesting effects. The algorithm turns on vehicle lights when there is heavy rain (Figure 15a). It changes the appearance of trees in snowy conditions (the leaves are removed as in winter) (Figure 15b). In addition, it is able to control the amount of snow on the road, by following the wear patterns on the road (Figure 15b). It is also capable of reproducing halo effects in fog (Figure 15c).

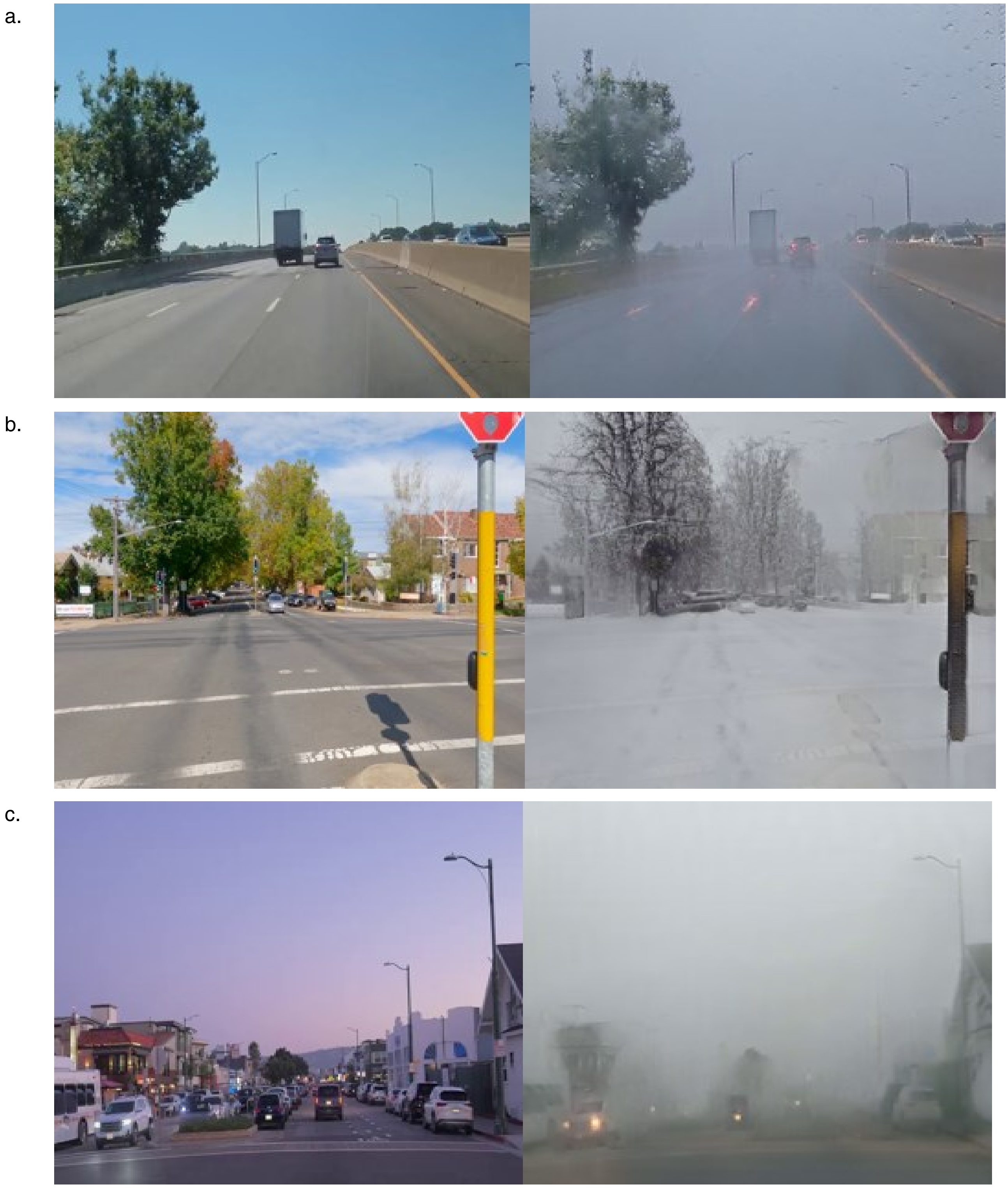


Figure 15: Example of interesting side effects learnt from the data-driven noise model.

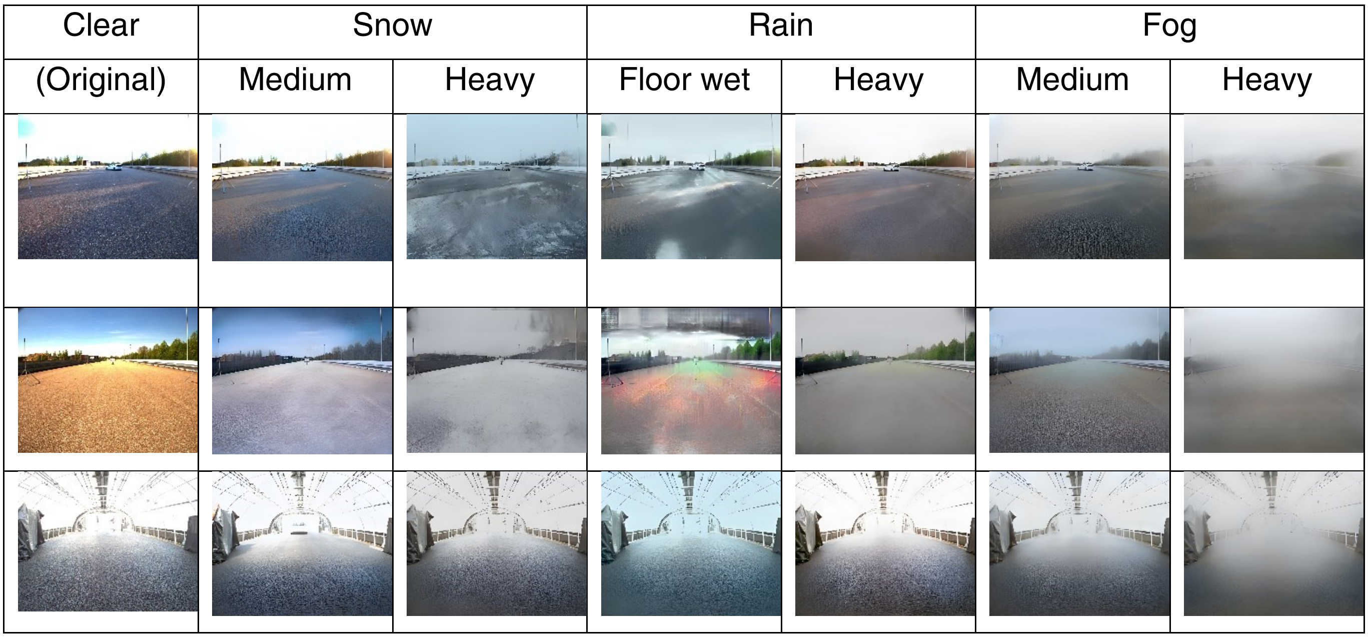


Figure 16: Results obtained by data-driven noise models on the REHEARSE database.

We also applied the models obtained to the images in the REHEARSE database (D3.2), as shown in Figure 16. As a reminder, this database will be used to validate the data-driven models in task 3.4. The following section discusses the camera noise models that have been developed.

2.6 Discussion

We attempted to simulate degraded weather conditions such as rain, snow, and fog using the CycleGAN technique and the Group2Group method. The data-driven noise models developed can generate fog, rain, and snow conditions, with 3 classes of intensity available. The model learning phase required the aggregation of new databases including several hundred thousand images. The major advantage of the data driven models developed is that they allow the primary effects linked to the weather to be considered, as well as all the induced effects (such as changes in vegetation, changes in surface conditions, etc.).

We obtained visually interesting results, although sometimes certain artifacts appear, as illustrated in Figure 17. These artifacts occur because we do not directly control which style elements the algorithm transfers from one group of images to another. Particularly when we have difficulties perfectly aligning the image subgroups, artifacts may manifest. Sometimes, even using the Group2Group method, it is challenging to achieve perfectly parallel subgroups where the only difference is the weather, which can also lead to artifacts. Moreover, although aligning the subgroups is not as complex as creating databases for Pix2Pix, it remains a challenge, as we observed during the simulation of fog.

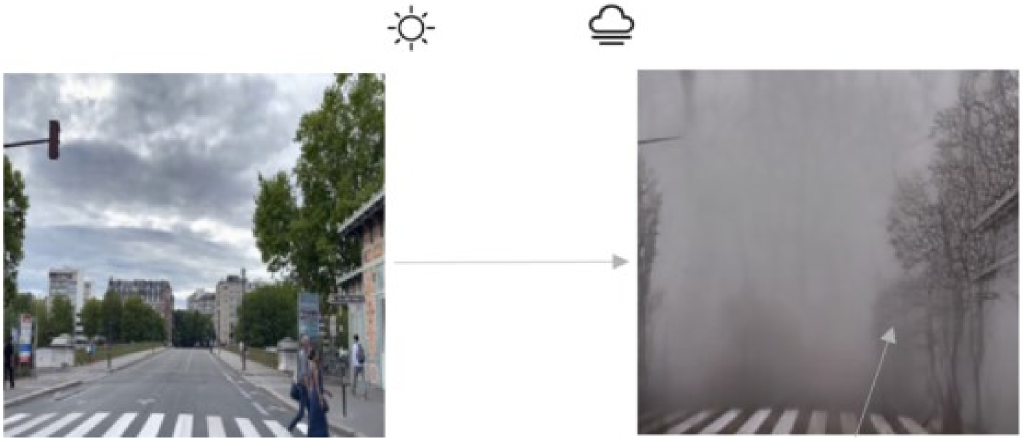


Figure 17: Some artefacts in simulated images

Ideally, we would be able to more precisely influence the style elements that the algorithm transfers, to focus only on those that are relevant for our task, namely those related to weather conditions. There are primarily two sub-types of styles:

1. **Styles focused on form and structure**: These styles concentrate on the representation of shapes, outlines, and textures. For example, in city, there are a lot of straight lines, and in forest, there are a lot of smooth curves.
2. **Styles focused on atmosphere**: These styles emphasize intangible elements such as light and colour.

The second type of style is particularly relevant to our work because it relates to the overall ambiance of an image. This classification helps us more precisely understand what defines the "style" of an image, distinguishing the influences of structure from those of the atmosphere. These distinctions are also useful for refining our approach to neural style transfer, allowing us to specifically target elements to manipulate or preserve in our models.

In the technique of neural style transfer, it is possible to extract style features from an image from the lower layers of a neural network, which are associated with details such as the shape of objects, while features extracted from the upper layers relate more to the general ambiance, including weather conditions. We plan to conduct tests on the layers of the CycleGAN discriminator to identify those that best capture elements related to weather conditions. This work should also help reduce artifacts. It will be carried out as part of Task 5.2, on the detection of weather conditions by camera.

3 LiDAR

3.1 Literature review

As described in the introduction, the modelling of degraded weather conditions can be physics based or data driven. Physical noise models are widespread [14] [15] [16] [17] [18] [19]. [15] provides a state-of-the-art review of these models and proposes some measurements that validate the developed physical models. Real measurements are taken in chambers designed to generate fog and rain. But this kind of validation against real data is too rare [17] [19]. Another limitation of this type of model can be the computation time. The proposed models are often complex and can induce high computation times. Finally, this type of model often focuses only on the atmospheric part of hydrometeors. For example, they do not consider the internal algorithms of real sensors, which are often not even known. Moreover, changes in the reflectivity of materials are not always considered (due to wetness or snow covering). For instance, most models do not account for the fact that the ground is wet [20], unlike [21] which does. Thus, physic-based models can be partial if all the weather-related elements are not modelled one by one.

The second part of the literature concerns data-driven models, also called empirical. This is the focus of our work, with physic-based models being the subject of task 3.4. Works on data-driven models are much less represented in the literature. Some studies only analyse the effects of weather on LiDARs without proposing an associated modelling [22] [23] [24] [25]. These studies concern the effects of fog and rain. In addition to most models, [26] [27] consider the spray phenomenon (water thrown by vehicle wheels) rather than the atmospheric part (falling raindrops). To our knowledge, only one reference proposes work similar to ours. It is a model based on a CycleGAN [28]. As this type of model requires a lot of training data, the learning was conducted on the LIBRE dataset [29]. This dataset is created in weather conditions simulated in rain and fog chambers. The main limitation is that the model is learned on unnatural data, unlike what we propose. Furthermore, the intensity levels are fixed and the authors report that they make significant errors for extreme weather conditions. Finally, the real-time aspect is not guaranteed with this type of model with current computing power. Our model takes a different approach by processing data ray by ray, as shown in the following section.

3.2 Model Structure

We propose a noise model to simulate the impact of weather on LiDAR data acquired in clear weather conditions. This model operates on a ray-by-ray (point-by-point) basis, meaning it does not process a point cloud as a whole. This approach, which diverges from the only similar works we have found in the literature [28], is inspired by physical phenomena. Indeed, for weather simulation, we are not dealing with a scenario where the geometry of the scene is important. For example, for segmentation, it is necessary to consider the point cloud as a whole to process it, due to the need to account for coherence between nearby points. Here, we are dealing with a phenomenon that impacts points indiscriminately from one another, as the weather is considered homogeneous within the LiDAR's acquisition space. Adopting this assumption allows us to overcome a major problem concerning data quantity: instead of having X point clouds (thus X elements for learning), we benefit from X x 26,700 elements (since there are about 26,700 points in a minute-long point cloud) for learning. This enables us, with real data collected over a winter, to have enough data to conduct the training. Another advantage of this point-by-point method is that it is less costly in terms of computation time and can be parallelized for real-time processing in the future. This method, which does not take account of geometric appearance, has a few disadvantages: scenarios involving localised spray behind vehicles will not be simulated; similarly, ground reflections (puddles or wet ground) are only partially addressed.

The method used is based on observations made in the state of the art and our data. Figure 18 presents, as an example, an average point cloud acquired in clear weather and an average point cloud acquired in foggy weather. The literature [29] [28] [25] notes that several phenomena are observed when LiDAR encounters degraded weather, and they are present on the Figure 18:

1. The most distant points or those with low reflectivities disappear. For example, in Figure 18a , it can be seen that the points between 50 and 60m in the 45° - 90° quarter are present in clear weather but disappear in fog.
2. The more intense the weather, the more points disappear. This is also observed in our data, although the figure does not show it. The classification results from part 3.4.3 will demonstrate this.
3. Some points remain unaffected by the weather. This is the case in Figure 18b, where it can be seen that between 6 and 10m, in the 280°-340° quarter, the points of the scene are present with or without fog.

4. A sparse "ghost" point cloud close to the LiDAR sometimes forms within the 3-10m range. This is shown in Figure 18b and occurs for intermediate fog. This cloud can be simply filtered because the reflectivity of the concerned points is low.
5. A dense ring of "wall" points forms just around the LiDAR <3m. This is shown in Figure 18b and occurs for denser fogs. This cloud can be simply filtered because it is abnormally close to the LiDAR (in a road context, targets less than 3m are not possible).

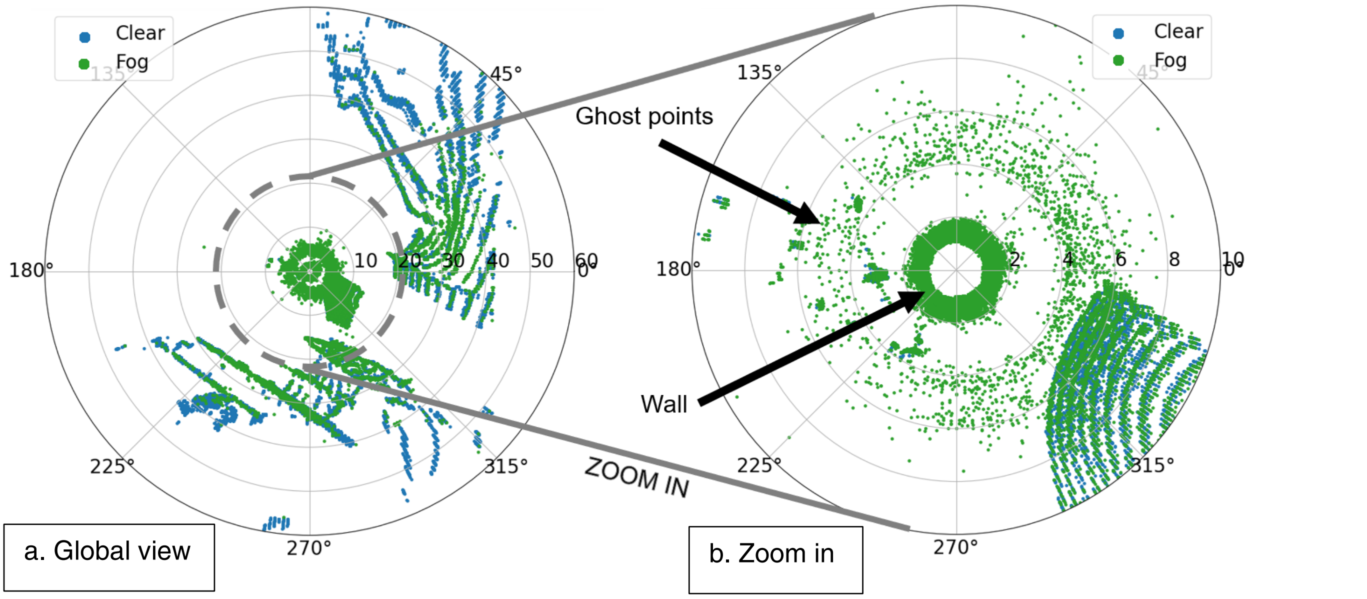


Figure 18: Example of a point cloud in clear and foggy conditions. Bird-Eye-View (BEV).

The proposed model thus processes each point in the point cloud individually. For each point, as shown in Figure 19, it first classifies the point as 'no change' or 'disappear'. To do this, we calculate the vanishing distance of the point, similarly to a previous study [25]. The calculation of the vanishing distance is detailed in section 3.4.3. The inputs for classification are therefore the distance, the reflectivity of the point (in clear weather), and the intensity of the weather. Note that the azimuth and vertical angle are not changed during the process. With this method, we achieve different behaviour for each of the two classes:

- 'No change': neither the distance nor the intensity of the point is changed.

- 'Disappear': the point is removed from the cloud.

It is then possible to add points of the type "Ghost point" or "Fake wall". However, although these have been characterized in section 3.4.5 for the LiDAR we used (VLP16), they are not added to the final model for two reasons: they are very easily filterable, and some LiDAR models natively filter them.

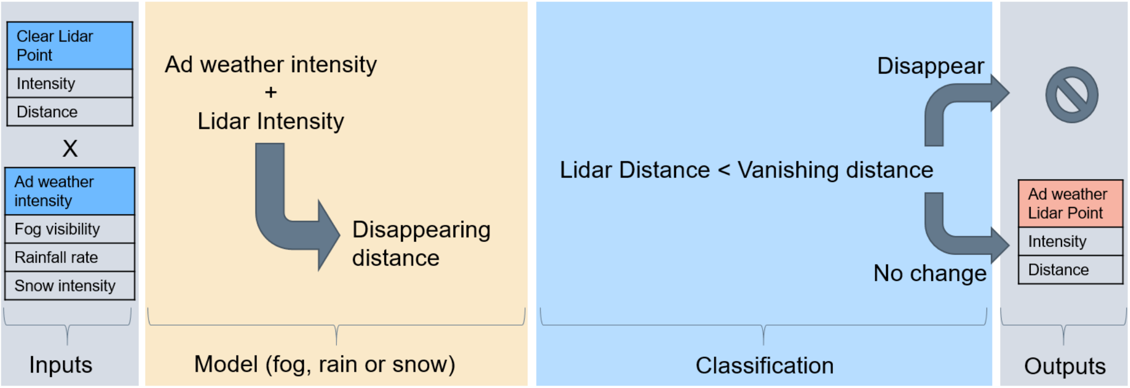


Figure 19: Diagram of the algorithm structure used to simulate degraded weather conditions on a LiDAR point. Note: this method is applied independently to each point of the complete point cloud.

This section has thus shown the method used to model degraded weather (fog, rain, snow) on LiDAR data acquired in favourable weather conditions. This data-driven method requires a significant amount of data for training. The following section shows what data are used.

3.3 Dataset

The database used for training was created by CE as part of the ROADVIEW project. It consists of data acquired at the summit of PDD, France. At the PDD site, see Figure 20, CE has installed camera, RADAR, and LiDAR sensors, as well as weather sensors. The installed weather sensors allow for the classification of weather (clear, fog, rain, snow) and for characterizing the intensity of the weather (MOR or rain intensity). Sensor data are collected at a rate of one measurement per hour. Each LiDAR measurement lasts one minute. The weather present during the sensor recordings is also recorded. For the LiDAR, data from the winter of 2022-2023 were used.

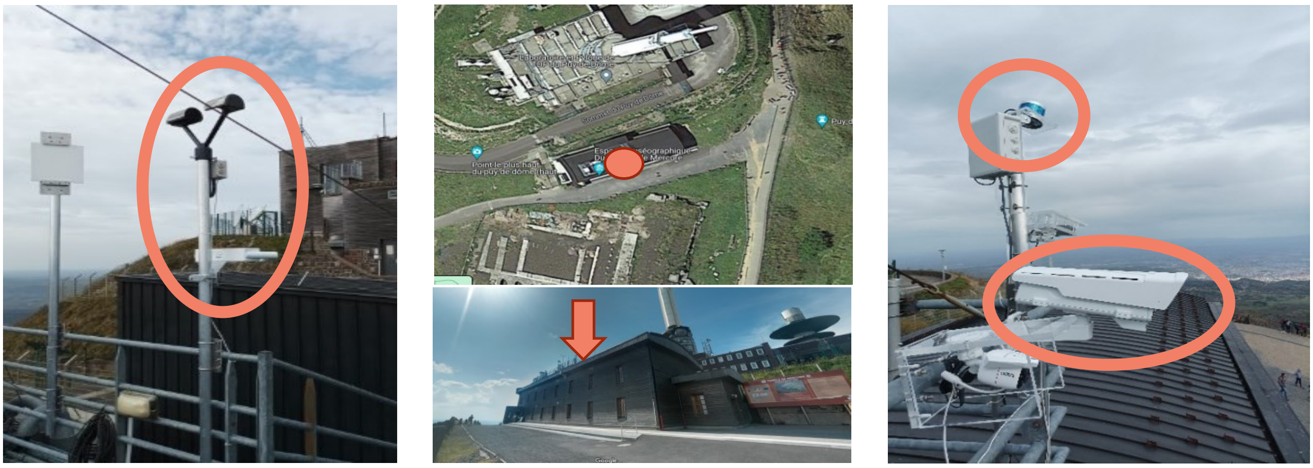


Figure 20: Presentation of the PDD site. The LiDAR is installed to capture surrounding objects and empty space. The presence of weather sensors installed by CE to characterize the weather during LiDAR recordings is also visible.

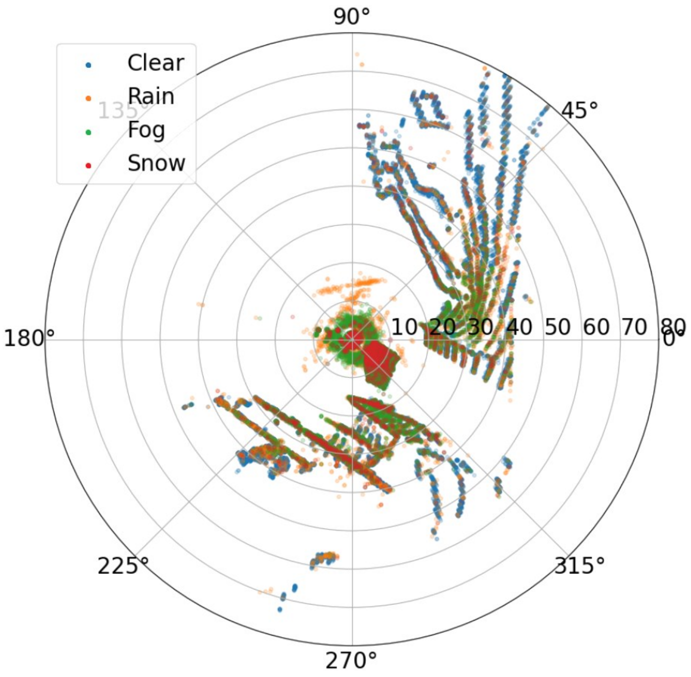


Figure 21: Average point cloud resulting from the measurement campaign, in BEV. It can be seen in the point cloud that in the presence of degraded weather, a phantom point cloud is created around the LiDAR, within a close disk with a radius of less than 8m.

For the winter of 2022-2023, the following weather sensors were used: Vaisala PWD12 (rain, snow, fog), Luft VS2K (fog), and Vaisala WXT530 (rain). This combination of sensors allows for redundancy in measurement to compare measurements and maintain continuity in case one of the sensors fails. Indeed, the site is difficult to access during winter, and maintenance is not always possible.

The database contains 2 723 LiDAR point clouds, each lasting one minute. For each point cloud, the weather is known. Therefore, we classify the point clouds to retain 4 groups: clear weather, fog, rain, and snow. Some point clouds are discarded because they do not fall into any of the 4 categories.

For clear weather, only point clouds with the following characteristics are kept:

PWD12_Weather Type = "Clear"
PWD12_Rain intensity (mm/h) = 0
WXT530_Rain intensity (mm/h) = 0
VS2K_MOR (m) >= 2000
PWD12_MOR (m) >= 2000

There is neither fog (MOR>0) nor rain (Rr=0) in the case of clear weather. Moreover, the weather sensor announces clear weather. For rain, only point clouds with the following characteristics are kept:

PWD12_Weather Type = "Rain"
PWD12_Rain intensity (mm/h) > 0

For rain, it is verified that the weather sensor announces a rain intensity and detects rain. For fog, only point clouds with the following characteristics are kept:

PWD12_Rain intensity (mm/h) = 0
WXT530_Rain intensity (mm/h) = 0
VS2K_MOR (m) < 400
PWD12_MOR (m) < 400
ABS(PWD12_MOR (m) - VS2K_MOR (m))<=60

For fog, the consistency between the two visibility sensors is checked, and it is ensured that there is a MOR below 400m. This value is chosen because it defines the presence of fog in road context according to standard NF P 99-320. For snow, it is ensured that the weather sensor announces the presence of snow.

PWD12_Weather Type = "Snow"

The conditions on snow are more lenient, as the occurrence is much less frequent.

After performing the classification using weather sensors, data from three cameras (in three directions) were used to visually validate that the data are correctly classified. This step allows for the removal of point clouds misclassified by the weather sensors. For clear weather, visual validation helps keep only point clouds where the ground is dry (all point clouds with wet ground are removed). Similarly, all weather conditions where the weather is overcast are removed to ensure ideal visibility conditions. Conversely, for rain, only images with rain and wet ground are kept. Images where snow is present are removed. Similarly, for snow, only the point cloud where there is actually snow in the atmosphere and on the ground is kept. Following this classification work into four weather groups, groups of point clouds are listed in Table 3. As shown in the table, visual filtering eliminates many cases where the weather is poorly measured by the sensors. Weather measurement at extreme sites like PDD is challenging, and visual validation is therefore important. In the end, there are 45 point clouds distributed into four classes. Thanks to our ray-by-ray based model, these data are sufficient to conduct training as shown in section 3.5. Before this, some technical details are given in the following section.

	Before video filter	After video filter
Clear	49	5
Rain	25	11
Fog	40	28
Snow	6	1

Table 3: Number of point clouds available in the database, by weather, for the winter 2022-2023.

3.4 Implementation details

3.4.1 General Approach

Firstly, it's important to note that one separate model is created for each weather condition, resulting in three LiDAR models (for rain, fog, and snow). All developments were performed in Python.

To carry out the training, clear weather data and degraded weather data were paired by direction (azimuth and vertical angle pairs). As shown in Figure 22, for each direction in clear weather, points in the same direction from other timestamps (e.g., during fog) were paired. This method thus allows for the creation of tuples (Clear Weather Distance, Clear Weather Reflectivity, Weather Intensity, Degraded Weather Distance, Degraded Weather Reflectivity), where Distance is the distance of the point detected by the LiDAR and Reflectivity is the reflectivity of the LiDAR point. These tuples with 5 values therefore contain 3 input data and 2 output data.

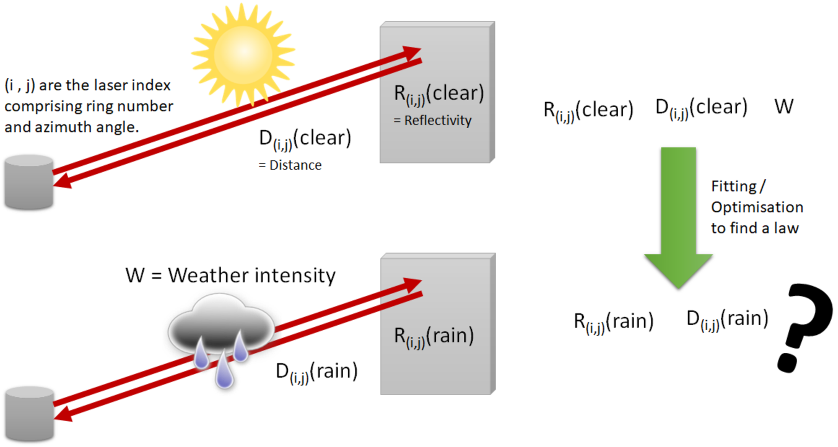

(i , j) are the laser index comprising ring number and azimuth angle.
$R_{(i,j)}$(clear)
= Reflectivity
$D_{(i,j)}$(clear)
= Distance
$R_{(i,j)}$(clear) $D_{(i,j)}$(clear) W
Fitting / Optimisation to find a law
W = Weather intensity
$R_{(i,j)}$(rain)
$D_{(i,j)}$(rain)
$R_{(i,j)}$(rain) $D_{(i,j)}$(rain)
?

Figure 22: General principle of the proposed data-driven model for LiDAR

Following this association of clear and degraded weather data, the points were classified according to the two classes defined in the method, as shown in Figure 23. The goal of the proposed method is to find the vanishing distance as a function of the weather intensity and the LiDAR reflectivity, as explained in section 3.4.3. Before presenting the results, it is important to better qualify the database from a weather perspective.

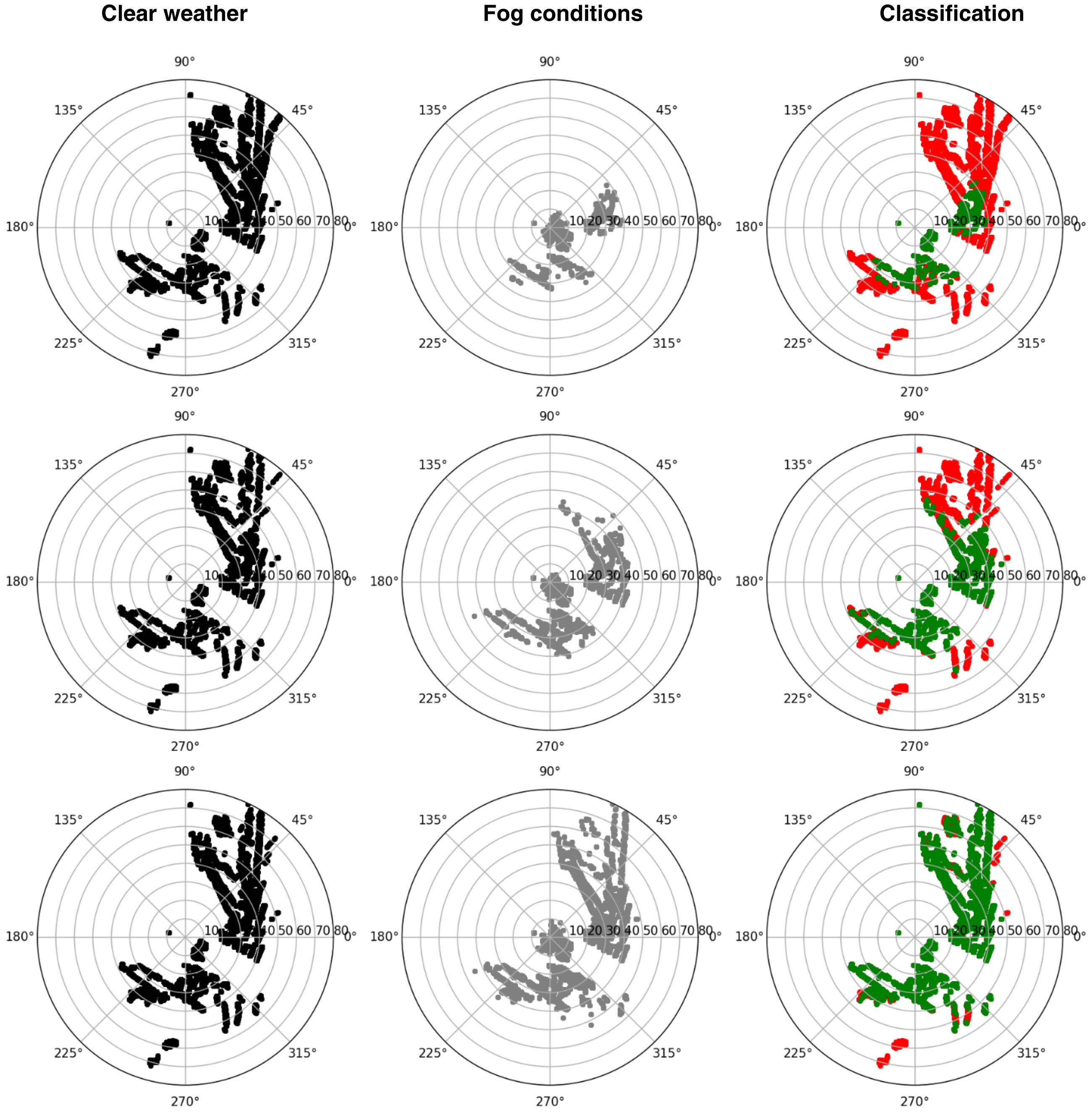


Figure 23: Classification of 'no change' or 'disappeared' points, for three levels of fog in BEV. MOR from top to bottom: 50-75m, 75-100m, 100m-200m. Black point cloud in clear weather, grey point cloud in fog, green point classified as 'no change' and red point.

3.4.2 Weather intensity definition for the three classes

The next step is to define a weather intensity for each of the three weather conditions. For calculating the vanishing distance, the MOR recorded during the event is used regardless of the weather class. Indeed, rain and snow are sometimes measured in resultant MOR [30], and it turns out that this measure correlates better with vanishing distance in our case [25]. From theory point of view, this is logical because the LiDAR's laser beam is attenuated by the weather, hence the MOR based on a transmission rate is perfectly correlated with this. Then, Figure 24 is showing the MOR distribution of the three classes in the database.

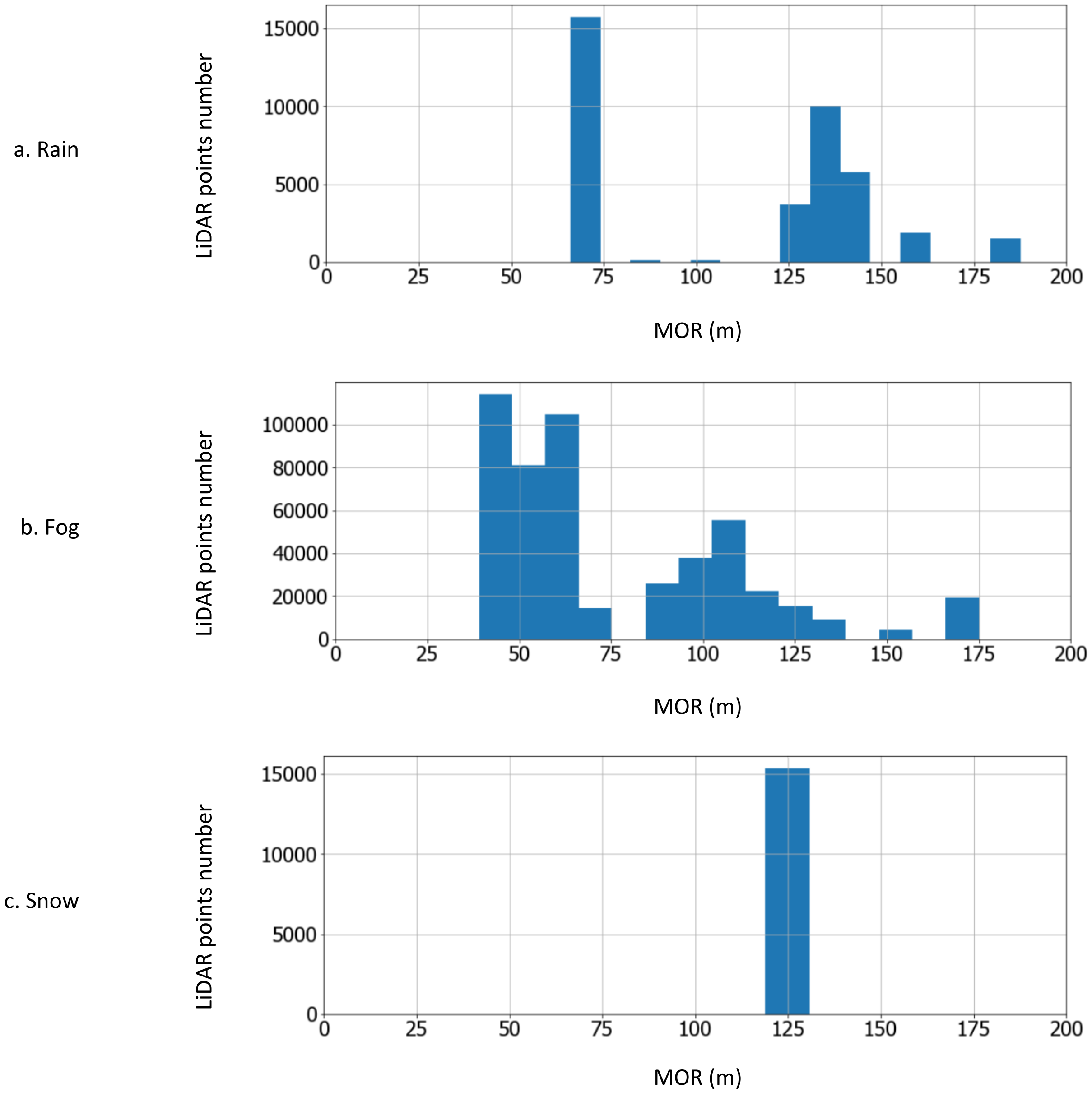


Figure 24: Histogram of MOR on the LiDAR training database (PDD winter 2022-2023).

It is observed from the graphs in Figure 24 that the data are well-distributed for fog and rain with values ranging from 50m to 175m. However, for snow, there is a lack of data to have something truly representative. The available database necessitates the use of a vanishing distance calculation model capable of extrapolating beyond the available data range. Indeed, fog validation trials are conducted on data ranging from 10 to 50m, yet there are no training data available in this range. The chosen model is therefore based on linear regression, which allows for this extrapolation, as will be explained in section 3.4.3.

In the case of rain and snow, the data available in the REHEARSE (D3.2) validation database is not MOR but rain intensity. For this, we propose converting rain intensity into MOR. This conversion is derived from measurements made within the PAVIN platform for creating the REHEARSE dataset. The conversion formula is as follows:

$$MOR = -0.8308 * Rainfall\ rate + 159.16$$

where MOR is expressed in m and rainfall rate in mm/h.

It is obtained by regression on the data from the REHEARSE validation database, as shown in Figure 25.

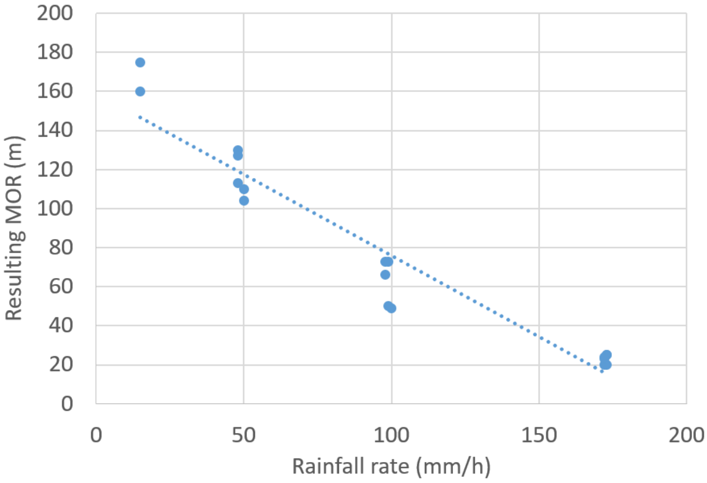


Figure 25: Conversion of rain intensity into equivalent MOR using data from the REHEARSE database.

3.4.3 Method to estimate Vanishing Distance

To estimate the vanishing distance, we propose plotting 'no change' and 'disappear' points on Figure 26. On this figure, we plot the distance of the LiDAR point according to the intensity of the weather, colouring it by its class ('no change' or 'disappeared'). From the figure, it is clear that there is a strong relationship between the intensity of the fog and the vanishing distance. The vanishing distance is represented on this figure by the boundary between the two sets of points 'no change' and 'disappeared'. To automate the detection of this boundary, we propose the following method:

- For each MOR, we retrieve the 0.99% quantile (to exclude outliers). These points are marked in black on Figure 26.

- We calculate the linear regression curve of the obtained points, resulting in the grey line on Figure 26. We set the interception of this line to 0 because theoretically, the vanishing distance is zero for an MOR of zero.

- Finally, we will use the slope of the obtained curve, which we will call the vanishing slope in the following, to find the vanishing distance from the MOR when using the model. This vanishing distance will be used to classify points, as explained in section 3.2.

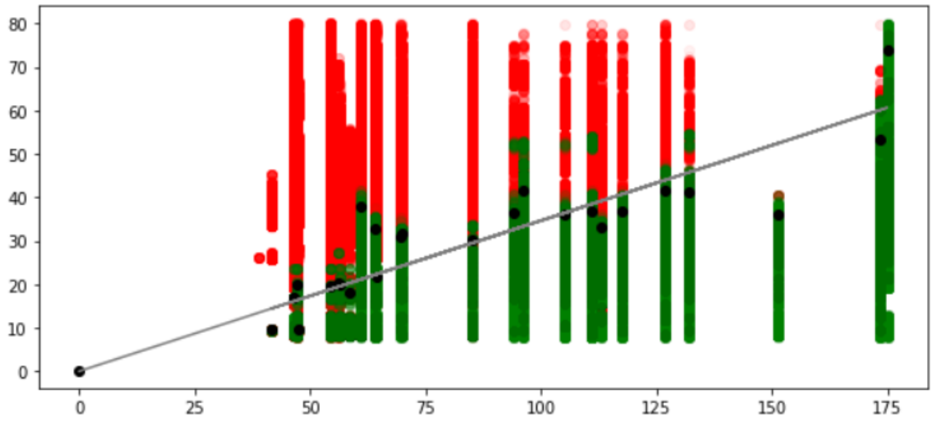


Figure 26: Method for determining the vanishing slope, on the fog class of the database.

3.4.4 Details on points reflectivity

It is interesting to consider whether the reflectivity of the LiDAR point plays a role in the vanishing distance of the point. Indeed, in the previous section, all point reflectivities were mixed. To verify that reflectivity does not impact, we therefore propose to check the vanishing slope using points with specific reflectivity. The method from the section 3.4.3 is therefore applied for points having reflectivities in the following intervals:

We then plot the vanishing slope according to the LiDAR reflectivity on the Figure 27. From this graph, we observe that this slope is constant for snow. However, for rain and fog, we note that the vanishing slope is lower for lower reflectivities (<10). A lower vanishing slope means that the vanishing distance is shorter and therefore that the weather phenomenon has more impact on the data. Thus, for lower reflectivities, rain and fog have more impact. For this, the vanishing slope will be modulated for the case of rain and fog for reflectivities < 10. For snow, the vanishing slope will be considered as a unique value regardless of the point's reflectivity. Table 4 therefore summarizes the vanishing slopes for different weather conditions.

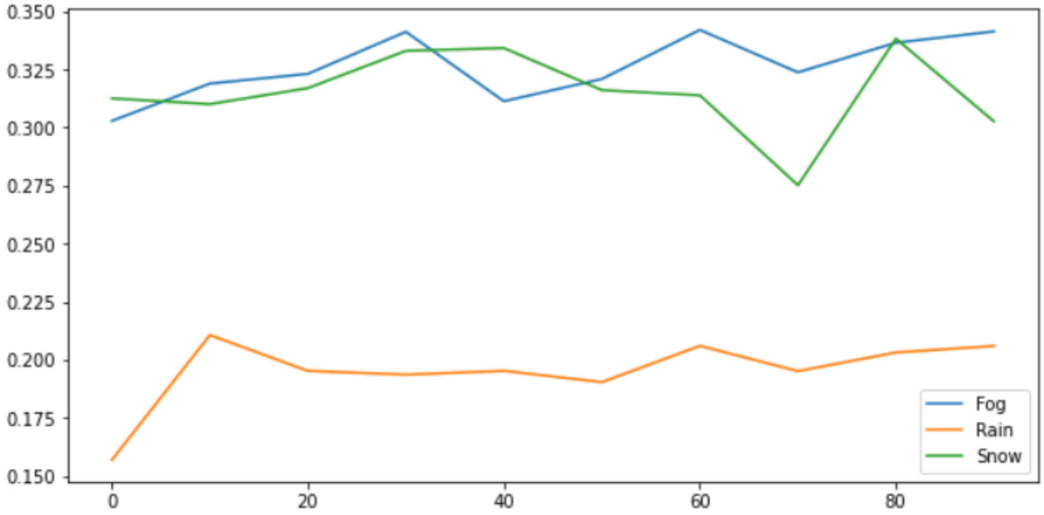


Figure 27: Vanishing slope according to the reflectivity of the LiDAR point.

	Vanishing slope	
	Point reflectivity < 10	**Point reflectivity ≥ 10**
Fog	0.30286536	0.34634364
Rain	0.15688675	0.22530445
Snow	0.32685039	0.32685039

Table 4: Vanishing slope of the data-driven model for rain, fog, and snow.

From the vanishing slope, we calculate the vanishing distance following the equation:

Vanishing distance = Vanishing slope x MOR

And for rain and snow, we obtain the equivalent MOR through the equation given in section 3.4.2.

3.4.5 Effect of 'Ghost Points' and 'Fake Wall'

As mentioned in the introduction, the study of LiDAR has highlighted three phenomena: the vanishing of certain points, the appearance of ghost points (point cloud 'ghostPoint' on Figure 28), and the appearance of a ring / fake wall around the LiDAR (point cloud 'fakeWall' on Figure 28). The first phenomenon, the most significant, was treated and included in the model. The other two effects are characterized here. They were ultimately not added to the model because:

1. the number of points is not high,
2. they are not replicable across all LiDAR brands, and
3. they are easily filterable because they represent very low point reflectivities.

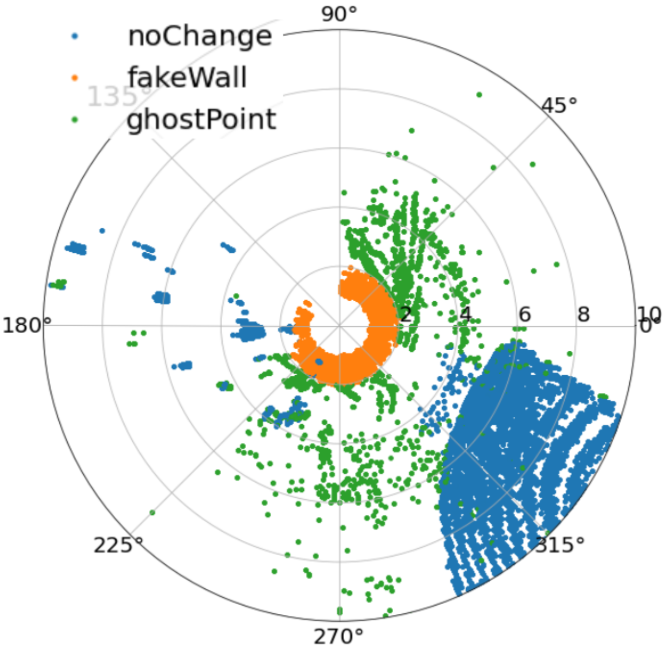
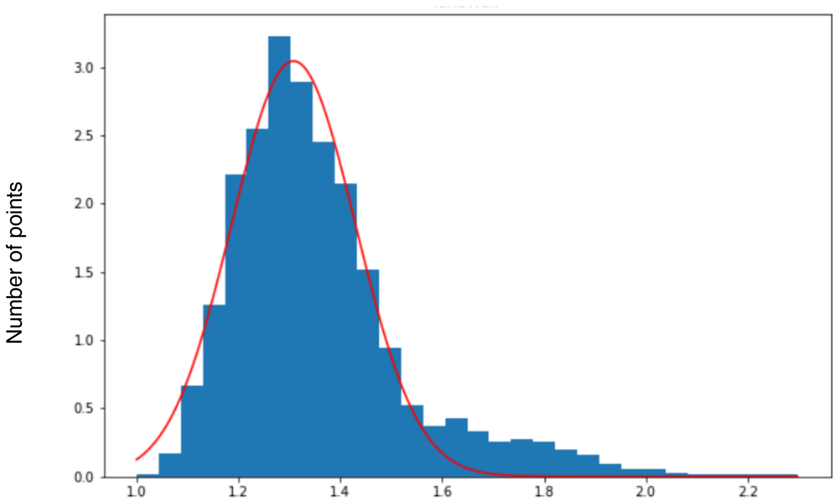


Figure 28: BEV view of the fog point cloud. The data have been classified using clear weather data into three classes.

In the course of the analysis, it was possible to characterize ghost and ring type points. The distribution in intensity and reflectivity of these points could be plotted, as shown in Figure 29 in the case of fog and the distance of the point. Then, on this distribution, we chose to fit a Gaussian law. The obtained parameters are summarized in Table 5.



Distance of the LiDAR point

Figure 29: Histogram of the distance of the points from the ghost point cloud, which appears in fog conditions (blue). On this histogram, optimization with a Gaussian law was performed (red).

		Fog		Rain		Snow	
		Distance	Reflectivity	Distance	Reflectivity	Distance	Reflectivity
Ring	**Mean**	1.308	1.461	1.282	1.912	1.549	2.688
	Std	0.172	3.055	0.212	3.119	0.380	6.077
Ghost	**Mean**	5.011	4.113	4.631	1.250	6.549	11.032
	Std	1.753	11.533	1.263	2.971	6.481	11.770

Table 5: Coefficients of Gaussian laws optimized on the phantom and ring point clouds in the presence of degraded weather.

Section 3.4 has presented the technical details of the implementation of the models. The following section therefore presents the simulation results obtained on some examples. As a reminder, the validation phase of the models is carried out within the framework of task T3.4, jointly with the physic-based noise models.

3.5 Results

3.5.1 Fog

The results for fog data are presented using the databases from PDD and REHEARSE. Figure 30 shows the results obtained on the PDD database. In this figure, the clear weather point cloud is represented in black, the simulated data in blue, and the real data in orange. The figure includes fog at different levels chosen for MORs of 150m, 90m, and 60m. From the figure, it is evident that the simulated fog has a similar impact to real fog, with an increasingly strong impact as the MOR decreases. Subsequently, the noise model is applied to the test data from the REHEARSE database (D3.2). Since the model training was conducted on the PDD base, the results might be biased. Furthermore, the LiDAR model differs between the PDD database and REHEARSE, so it is interesting to verify if the behaviour is generally consistent. The results obtained on the REHEARSE database are thus presented in Figure 31. In this figure, 3 MOR levels (10, 20, and 50m) are depicted, featuring a scene containing a pedestrian and a cyclist at 15m from the LiDAR at the CE's PAVIN platform. Again, it can be observed in this figure that the effect of the simulated fog is similar to real fog. This also shows that the classifier based on a linear threshold retained allows extrapolation beyond the limits of the training database. Indeed, the strongest fog in the PDD database does not go below 50m MOR, whereas the data from the REHEARSE database (D3.2) are below that. Initial trials with more complex classifiers had shown significant limitations when extrapolating beyond the limits of the database. These visual results will be confirmed by a validation within the framework of task 3.4 of the project.

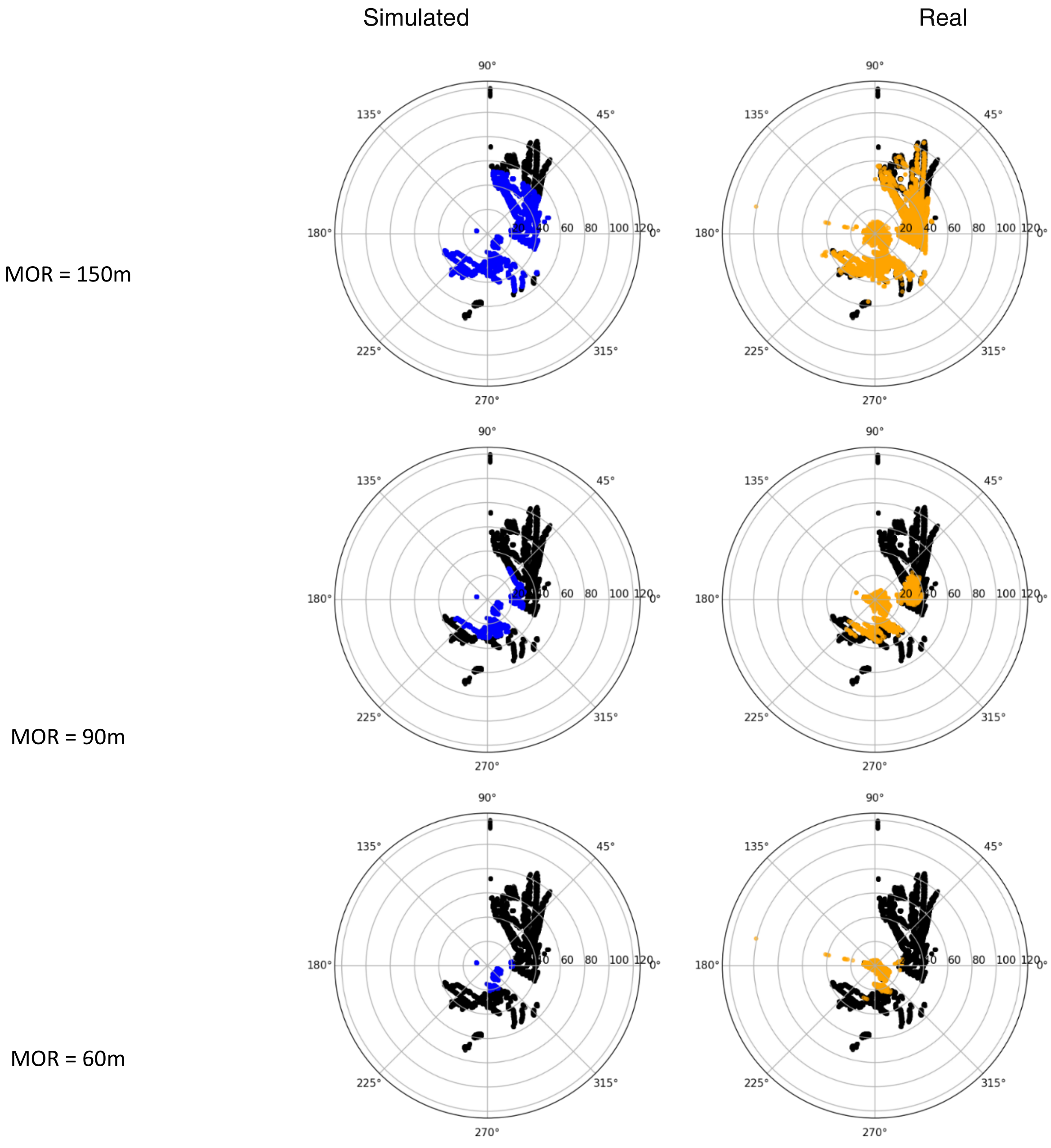


Figure 30: Results obtained by the fog data-driven noise model on LiDAR data of the PDD database, BEV. In black, the initial clear weather data; in blue, the simulated data with rain; in orange, the real data.

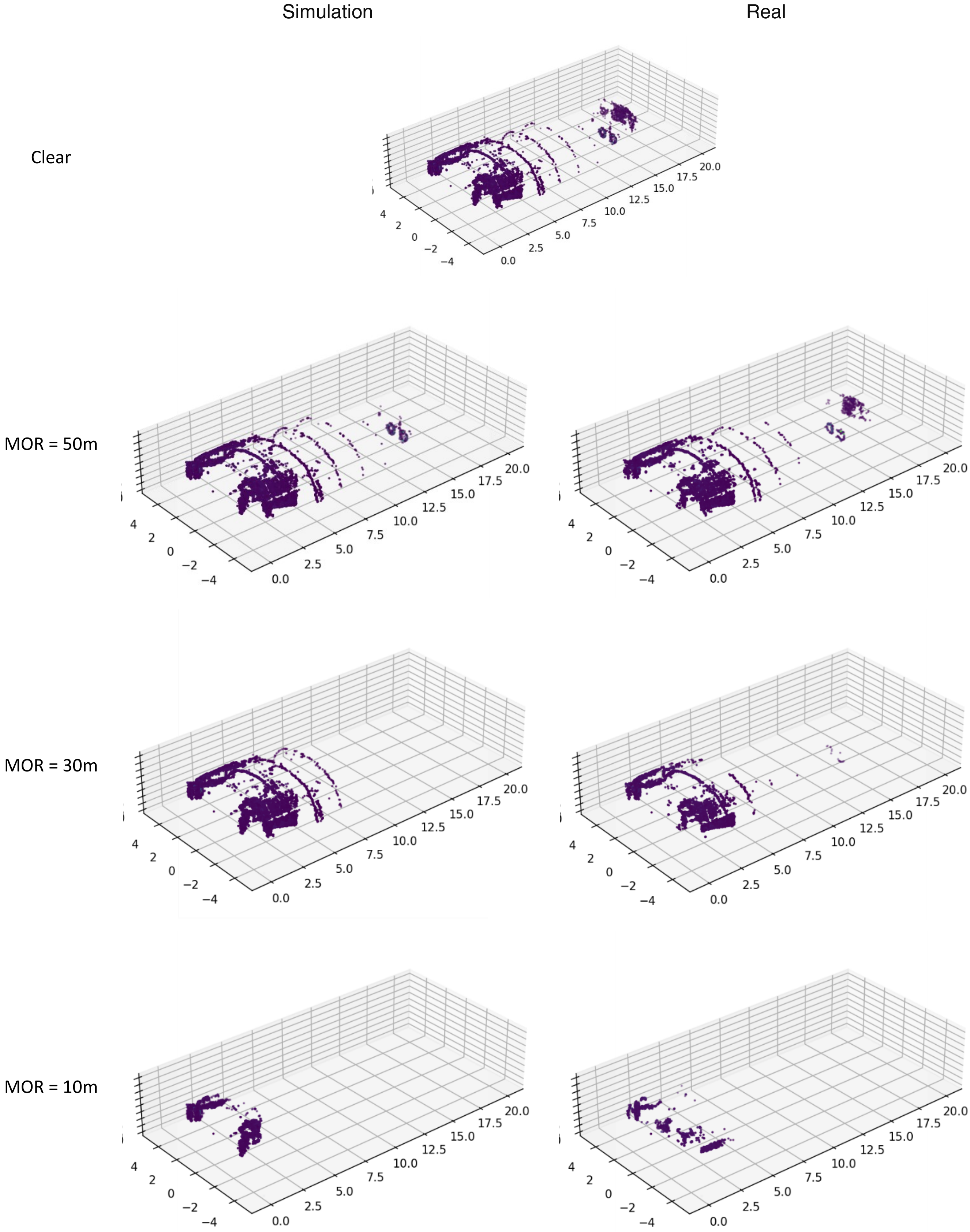


Figure 31: Results obtained by the fog data-driven noise model on LiDAR (Ouster) data of the REHEARSE database. Left is the simulation output, right is the real.

3.5.2 Rain

As with fog, a verification of the results obtained for rain was also carried out. Thus, Figure 32 shows the simulation results compared to real data from the PDD database. As with fog, it is evident that the model is capable of reproducing different levels of rain. Similarly, it appears visually that the results obtained from the REHEARSE database are consistent, as shown in Figure 33. On Figure 33, elements are present at a distance of 5m. These are the poles used to generate artificial rain. These initial visual results will be confirmed again, as for the fog, by a validation as part of task 3.4.

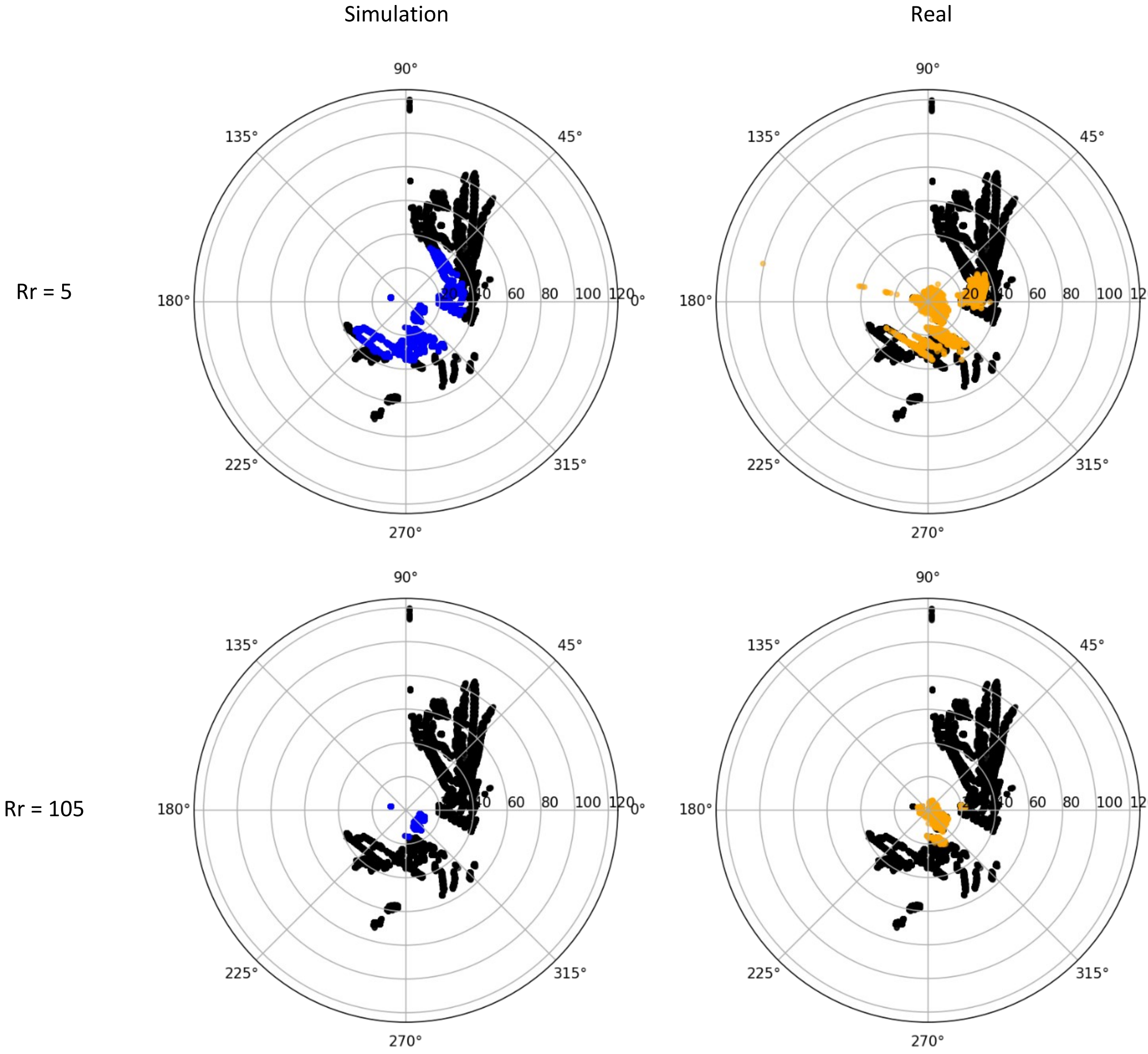


Figure 32: Model result on data from PDD, BEV. In black, the initial clear weather data; in blue, the simulated data with rain; in orange, the real data. Top line Rr = 5mm/h, bottom line Rr = 105mm/h.

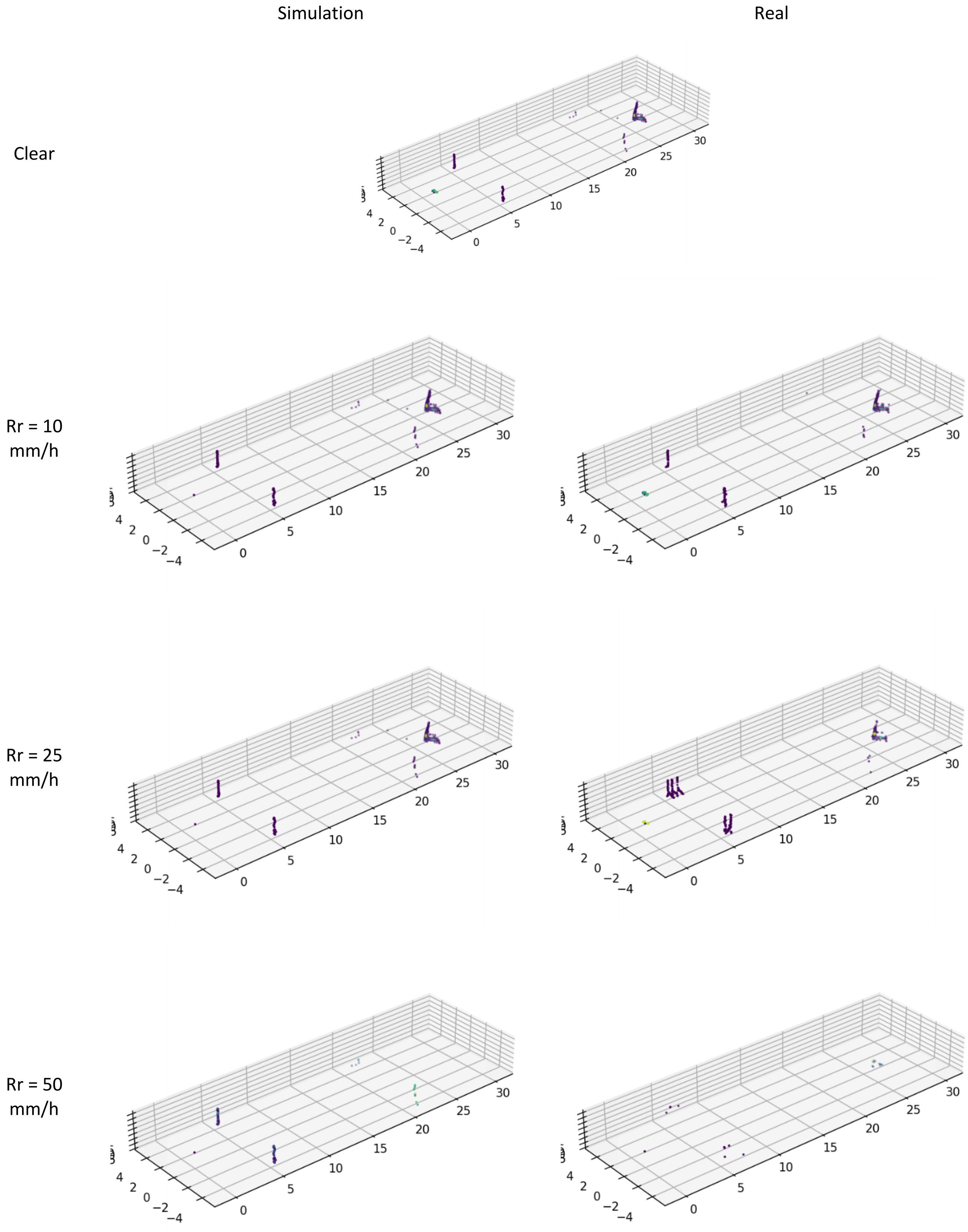


Figure 33: Results obtained by the rain data-driven noise model on LiDAR (Ouster) data of the REHEARSE database. Left is simulation output, right is real.

3.5.3 Snow

As with rain and fog, real and simulated data have been compared on the PDD database. The results obtained are shown in Figure 34. They show that the model corresponds well to reality. As a reminder, the snow conditions in the REHEARSE database are those acquired at PDD, hence the presence of a single figure in the case of snow.

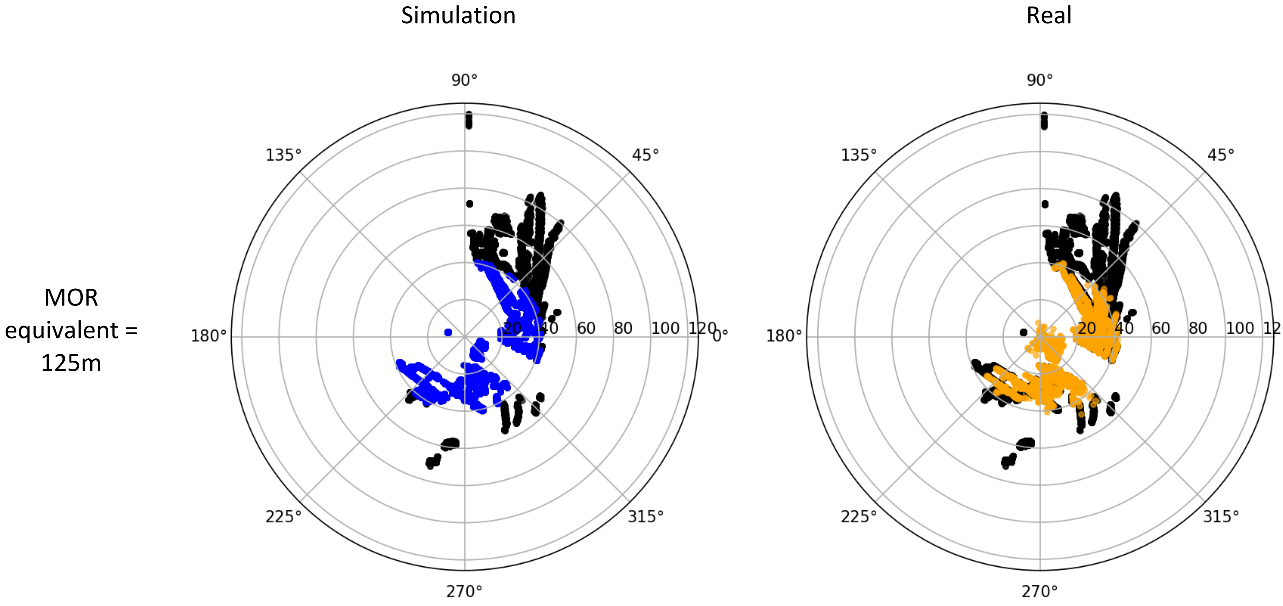


Figure 34: Results obtained by the snow data-driven noise model on LiDAR data of the PDD database, BEV. In black, the initial clear weather data; in blue, the simulated data with rain; in orange, the real data.

3.6 Discussion

A new data-driven noise model has been proposed for LiDAR. This model is based on a point-by-point classification of the point cloud. This classification is based on the estimation of the vanishing distance, already identified in the literature [25] but never published as a model to our knowledge. This estimation was carried out on real fog, rain, and snow data, thanks to the collection of a new database at the PDD site (France). Visual comparisons of the simulation results obtained with real data show that the model appears to be in line with reality. These comparisons were carried out on data collected at PDD, but also on data from the REHEARSE database, developed as part of task 3.2. More detailed validation measurements will be carried out on the REHEARSE database as part of task 3.4.

The advantages of the proposed method are numerous. First of all, it can be parallelized, and processing is not very time-consuming. This means that processing can be carried out in real time. Moreover, the training was carried out on natural data, which ensures that it is properly representative. Using a data-driven noise model also means that both the direct and indirect effects of weather change can be considered. For example, changes in surface properties are considered, as are atmospheric effects. Finally, the choice was made to carry out the training on one database (PDD) and the validation on a completely different database (REHEARSE). This bold choice is a guarantee of high validation quality, as there will be no possible bias effect. The limitation of the proposed model is that the training was carried out on a specific LiDAR model (VLP16). It could be interesting in future work to check the variability of the coefficients obtained on different LiDAR models.

4 RADAR

4.1 Literature review

Unlike LiDAR and cameras, which were discussed in the previous sections, the literature shows that RADAR is not or only slightly sensitive to degraded weather [31] [32]. As a result, very little work has been done on RADAR versus degraded weather conditions. As an example, [33] refers to 25 datasets containing RADAR data over the last ten years and only 2 of these ten show degraded weather conditions to our knowledge [34] [35]. As with other sensors, physical models and experiments do exist [33]. From a physical model point of view, the RADAR is only affected by heavy rain or heavy snow [36] [33]. [36] estimates from simulations that the detection range of RADAR can be reduced by 45% under severe rainfall conditions (150 mm/h), for a target such as a pedestrian, which is already tricky for RADAR under normal circumstances. This very high intensity of rain is, however, very rare in a concrete case, it corresponds to a huge storm peak, in which an automated vehicle would in any case have to come to an emergency stop. Limiting ourselves to extreme but realistic rain intensities for driving, the effect of rain does not exceed 20% for a pedestrian, and 8% for a car [36]. Regarding the impact of rain vs. snow, the literature is divided [31] with some explaining that the water content of snow has an impact on the severity on the RADAR, while others explain that the effect is identical between rain and snow. Concerning the experiments, [37] explains that the RADAR is not impacted by light rain and fog, that it is negligibly impacted by heavy rain and that it is only slightly impacted by snow. [38] [39] compares RADAR, LiDAR, and camera performances in simulated and real-world adverse climatic environments and concludes that RADAR outperforms LiDAR and cameras under the influence of rain.

With the work of the ROADVIEW project, and the new database that has been created at PDD with the 4D ZF ProWave RADAR, the following section will show that the RADAR is effectively not affected by the usual degraded weather.

4.2 Dataset

A specific database has been set up to create a RADAR model. This database was recorded at the PDD site during the winter of 2023-2024. As shown in Figure 35, the ZF 4D RADAR was installed to complement the sensor suite for the winter of 2022-2023: a Velodyne VLP16 LiDAR, 4 webcams and a Canon Axis Q1656-LE camera. For weather measurement, the sensor suite for the winter 2023-2024 period was: OTT Parsivel (Rain/Snow), Vaisala PWD12 (Temperature, Fog MOR, Rain/Snow), WXT530 (Rain/snow), and VS2K (Fog MOR).

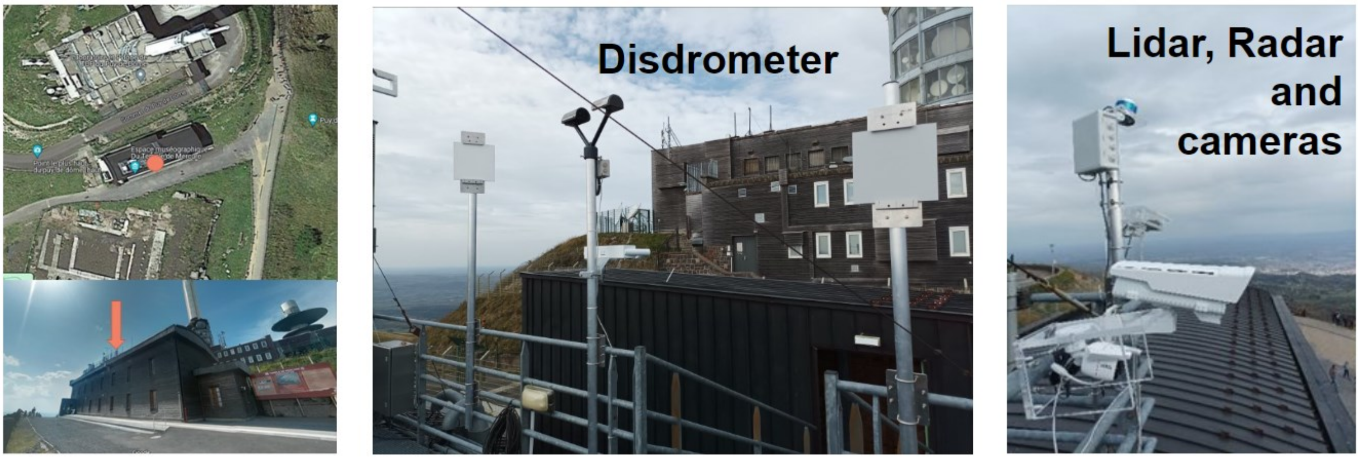


Figure 35: Sensors installed at the summit of the PDD for the RADAR measurement campaign (winter 2023-2024)

During this test campaign, one measurement is taken per hour, with 1 min duration for RADAR, 1 min duration for LiDAR, and 1 snapshot with each camera. Data is classified automatically using the following criteria. For clear weather, only point clouds with the following characteristics are kept:

$$\begin{gathered} \textit{Parsivel\_Rain intensity (mm/h)} = 0 \\ \textit{WXT530\_Rain intensity (mm/h)} = 0 \\ \textit{VS2K\_MOR (m)} \geq 2000 \\ \textit{PWD12\_MOR (m)} \geq 2000 \end{gathered}$$

There is neither fog (MOR>0) nor rain (Rr=0) in the case of clear weather. For rain, only point clouds with the following characteristics are kept:

WXT530_Rain intensity (mm/h) > 0
Parsivel_Rain intensity (mm/h) > 0
Parsivel_Weather Type ≠ "Snow"

For rain, it is verified that the weather sensor announces a rain intensity and detects that there is no snow. For fog, only point clouds with the following characteristics are kept:

Parsivel*_Rain intensity (mm/h) = 0*
WXT530_Rain intensity (mm/h) = 0
VS2K_MOR (m) < 400
PWD12_MOR (m) < 400

For fog, it is ensured that there is a MOR below 400m. This value is chosen because it defines the presence of fog in road context according to standard NF P 99-320. For snow, it is ensured that the weather sensor announces the presence of snow.

Parsivel_Weather Type = "Snow"
PWD12_Temperature < 5°C

The conditions on snow are more lenient, as the occurrence is much less frequent.

After performing the classification using weather sensors, cameras were used to validate that the data are correctly classified, as described in section 2.3. Following this classification work into four weather groups, groups of RADAR point clouds are listed in Table 6. There has been a significant reduction in the number of data points because a number of visual filters have been put in place. Data has therefore been removed in the following cases: when the ground does not correspond to the weather (wet ground during snow or snowy ground during rain), when the cameras were too obstructed, or at night. The following section presents the analyses and results obtained from this database.

	Before video filter	After video filter
Clear	341	74
Rain	67	8
Fog	140	18
Snow	157	4

Table 6: Number of point clouds available in the database, by weather, for the winter 2023-2024.

4.3 Results

As the literature shows that RADAR is very little affected by degraded weather, we decided to check this before setting up a model. The method presented in section 3.2 was reused on the RADAR data. As a reminder, this consists of classifying the RADAR points into 'no change' and 'disappeared' classes. It is then possible to check the rate of 'no change' points and the impact of the weather on the RADAR. Table 7 therefore shows the 'no change' rate for LiDAR (using the data and procedure in section 3.4.1) and RADAR. The results are clear. While LiDAR is strongly impacted by weather, in particular for fog, RADAR seems insensitive to degraded weather. As Table 7 shows there are less than 5% of points impacted, whether it is rain, fog or snow condition. However, the uncertainty of the measurement is of the order of 3% according to the no-change rates obtained in clear weather. It is therefore possible to conclude that the RADAR is insensitive to the degraded weather conditions encountered in the database. It is therefore important to better characterize the weather conditions present into the database in terms of weather intensity in order to qualify the limits of this result.

	LiDAR	RADAR
Clear	0.99	0.97
Rain	0.85	0.97
Fog	0.38	0.95
Snow	0.70	0.99

Table 7: 'No change' rate for LiDAR and RADAR, for each weather condition.

Figure 36 shows the histograms of weather intensity for the three weather conditions in the database. It can be seen from the figure that for rain and snow conditions, the intensities are fairly low with a maximum of 3mm/h. This corresponds to moderate rain according to the NF P 99-320 road standard. For fog, the MOR goes up to 50m, which still corresponds to fairly dense fog. The results obtained on the absence of impact from rain or snow are therefore in agreement with the literature for the intensities present in the database. Similarly, the result on the absence of impact of fog, whatever its density, is in line with the literature.

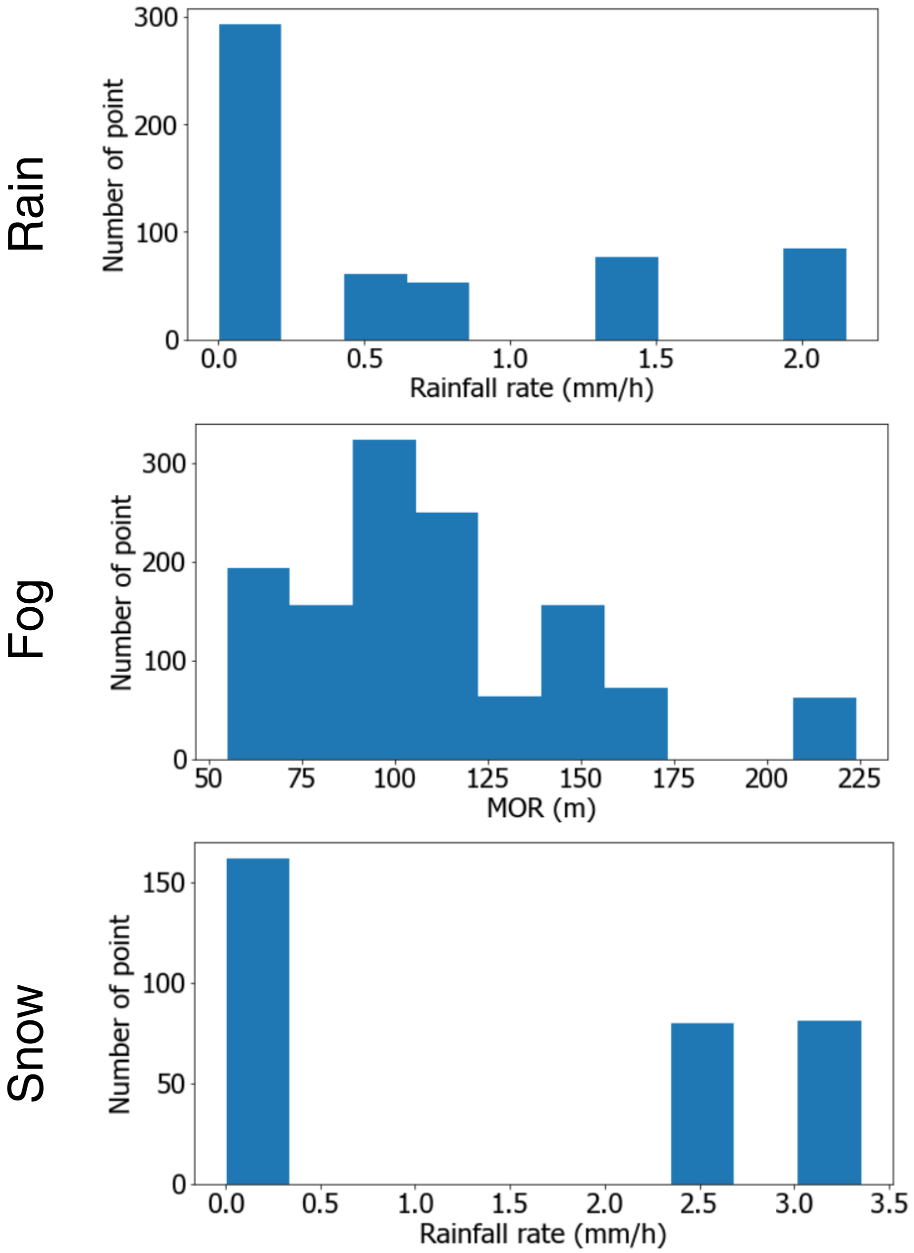


Figure 36: Histogram of weather intensities encountered in the database

4.4 Discussion

As far as RADAR is concerned, then, the weather has no impact according to our data. However, our database is limited when it comes to rain and snow, with intensities of up to 3mm/h. It is not possible with such data to extrapolate further, as has been the case with LiDAR. Other measurement campaigns should therefore be carried out to complete these results. However, the results obtained are in line with the literature on the field of study under consideration.

The conclusion is that the RADAR is insensitive to the weather, whatever the fog, and for precipitation intensities of up to at least 3mm/h for rain and snow.

5 Conclusions

This report presents how the data-driven noise models of the ROADVIEW project were produced. These noise models are published on the project's official GitHub[2]. They include models for cameras, LiDARs, and RADARs. They deal with fog, rain, and snow weather conditions. The choice was made to use different methods for each type of sensor in order to be better adapted to each data structure. For cameras, the Cycle-GAN model was chosen, with a new learning method called Group2Group. For LiDARs, a point-by-point processing method for the point cloud and vanishing distance estimation was used. Finally, it was shown that RADAR can be considered insensitive to current weather, in line with the literature.

The learning phase for the various models was carried out on new data, collected specifically by CE as part of task 3.3. These data were acquired in real conditions with natural degraded weather conditions. This aspect is very important, as it enables us to obtain models that are as natural as possible.

After training, the noise models obtained were compared visually with real data. This comparison was carried out both on the databases collected by CE and on the REHEARSE database (created as part of task 3.2 - D3.2). The precise validation of the models, using dedicated metrics, will then be carried out as part of task 3.4 with data from the REHEARSE database. The choice was made to carry out the training on CE databases and the validation on a completely different database (REHEARSE). This choice is a good practice as there will be no possible bias effect.

The advantage of the models obtained is that they can be used to specify a level of weather intensity. They can also be parallelized, which means they can be used in real time. Data-driven models have a major advantage over physic-based noise models. Thanks to the learning phase, they take direct account of all the impacts associated with the weather (disturbance in the atmosphere, changes in the state of surfaces, other secondary parameters). This avoids the need to add layers to model each aspect theoretically, as would be the case with physics-based models.

However, the models proposed here do have a few limitations. Firstly, the camera model only proposes weather intensity classes, and does not offer the possibility of choosing an exact intensity. In contrast, the LiDAR model allows you to specify a precise weather intensity. Secondly, data-driven models behave like black boxes, and there can sometimes be unwanted artefact effects. Finally, data-driven models are highly dependent on the database used, so it is difficult to estimate whether the results obtained are generalizable. In addition, finding metrics to tell whether a simulation output is realistic or not is a tricky subject.

This task feeds forward to T3.4 which will be to propose a validation of the physic based and data driven models. This will enable a proper comparison of these two different and complementary approaches.

[2] https://github.com/roadview-project/data-driven-noise-models

6 References

[1] L. A. Gatys, A. S. Ecker and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[2] P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, “Image-to-image translation with conditional adversarial networks(1 paired supervised),” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[3] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” 2017.

[4] T. Park, A. A. Efros, R. Zhang and J.-Y. Zhu, “Contrastive learning for unpaired image-to-image translation,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, 2020.

[5] X. Li, K. Kou and B. Zhao, “Weather GAN: Multi-domain weather translation using generative adversarial networks,” *arXiv preprint arXiv:2103.05422,* 2021.

[6] Y. Zhang, N. Huang, F. Tang, H. Huang, C. Ma, W. Dong and C. Xu, “Inversion-based style transfer with diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[7] C. H. Wu and F. De la Torre, “Unifying Diffusion Models' Latent Space, with Applications to CycleDiffusion and Guidance,” *arXiv preprint arXiv:2210.05559,* 2022.

[8] S. Yang, H. Hwang and J. C. Ye, “Zero-shot contrastive loss for text-guided diffusion image style transfer,” *arXiv preprint arXiv:2303.08622,* 2023.

[9] Y. Jia, L. Hoyer, S. Huang, T. Wang, L. Van Gool, K. Schindler and A. Obukhov, “DGInStyle: Domain-Generalizable Semantic Segmentation with Image Diffusion Models and Stylized Semantic Control,” *arXiv preprint arXiv:2312.03048,* 2023.

[10] S. Zheng, C. Lu and S. G. Narasimhan, *TPSeNCE: Towards Artifact-Free Realistic Rain Generation for Deraining and Object Detection in Rain,* 2023.

[11] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang and S. Paul Smolley, “Least squares generative adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017.

[12] W.-T. Chu, X.-Y. Zheng and D.-S. Ding, “Image2weather: A large-scale image dataset for weather property estimation,” in *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*, 2016.

[13] J. C. V. Guerra, Z. Khanam, S. Ehsan, R. Stolkin and K. McDonald-Maier, “Weather Classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of Convolutional Neural Networks,” in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 2018.

[14] D. Yang, Z. Liu, W. Jiang, G. Yan, X. Gao, B. Shi, S. Liu and X. Cai, “Realistic Rainy Weather Simulation for LiDARs in CARLA Simulator,” December 2023.

[15] A. Haider, M. Pigniczki, S. Koyama, M. H. Köhler, L. Haas, M. Fink, M. Schardt, K. Nagase, T. Zeh, A. Eryildirim, T. Poguntke, H. Inoue, M. Jakobi and A. W. Koch, “A Methodology to Model the Rain and Fog Effect on the Performance of Automotive LiDAR Sensors,” *Sensors,* vol. 23, p. 6891, August 2023.

[16] M. Bijelic, T. Gruber and W. Ritter, “A Benchmark for Lidar Sensors in Fog: Is Detection Breaking Down?,” *IEEE Intelligent Vehicles Symposium, Proceedings,* Vols. 2018-June, no. Iv, pp. 760-767, 2018.

[17] C. Goodin, D. Carruth, M. Doude and C. Hudson, “Predicting the Influence of Rain on LIDAR in ADAS,” *Electronics, MDPI,* 2019.

[18] 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,” July 2021.

[19] M. Hahner, C. Sakaridis, D. Dai and L. V. Gool, “Fog Simulation on Real LiDAR Point Cloudsfor 3D Object Detection in Adverse Weather,” in *IEEE International Conference on Computer Vision (ICCV 2021)*, 2021.

[20] M. Dreissig, D. Scheuble, F. Piewak and J. Boedecker, “Survey on LiDAR Perception in Adverse Weather Conditions,” April 2023.

[21] M. Hahner, C. Sakaridis, M. Bijelic, F. Heide, F. Yu, D. Dai and L. V. Gool, “LiDAR Snowfall Simulation for Robust 3D Object Detection,” in *CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022*, 2022.

[22] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter and W. Stork, “Weather Influence and Classification with Automotive Lidar Sensors,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[23] K. Montalban, C. Reymann, D. Atchuthan, P.-E. Dupouy, N. Riviere and S. Lacroix, “A Quantitative Analysis of Point Clouds from Automotive Lidars Exposed to Artificial Rain and Fog,” *Atmosphere,* vol. 12, 2021.

[24] R. H. Rasshofer, M. Spies and H. Spies, “Influcences of weather phenomena on automotive laser radar systems,” *Advance in Radio Science,* pp. 49-60, 2011.

[25] Y. Li, P. Duthon, M. Colomb and J. Ibanez-Guzman, “What happens for a ToF LiDAR in fog?,” *Transactions on Intelligent Transportation Systems,* 2020.

[26] C. Linnhoff, D. Scheuble, M. Bijelic, L. Elster, P. Rosenberger, W. Ritter, D. Dai and H. Winner, “Simulating Road Spray Effects in Automotive Lidar Sensor Models,” December 2022.

[27] J. R. Vargas Rivero, T. Gerbich, B. Buschardt and J. Chen, “Data Augmentation of Automotive LIDAR Point Clouds under Adverse Weather Situations,” *Sensors,* vol. 21, p. 4503, June 2021.

[28] J. Lee, D. Shiotsuka, T. Nishimori, K. Nakao and S. Kamijo, “GAN-Based LiDAR Translation between Sunny and Adverse Weather for Autonomous Driving and Driving Simulation,” *Sensors,* vol. 22, p. 5287, July 2022.

[29] A. Carballo, J. Lambert, A. Monrroy-Cano, D. R. Wong, P. Narksri, Y. Kitsukawa, E. Takeuchi, S. Kato and K. Takeda, “LIBRE: The Multiple 3D LiDAR Dataset,” March 2020.

[30] M. Colomb, P. Duthon and S. Laukkanen, *Deliverable D 2.1 : Characteristics of Adverse Weather Conditions,* 2017, p. 73.

[31] A. S. Mohammed, A. Amamou, F. K. Ayevide, S. Kelouwani, K. Agbossou and N. Zioui, “The Perception System of Intelligent Ground Vehicles in All Weather Conditions: A Systematic Literature Review,” *Sensors, MDPI,* 2020.

[32] X. Gao, S. Roy, G. Xing and S. Jin, “Perception Through 2D-MIMO FMCW Automotive Radar Under Adverse Weather,” in *2021 IEEE International Conference on Autonomous Systems (ICAS)*, 2021.

[33] Y. Zhou, L. Liu, H. Zhao, M. Lopez-Benitez, L. Yu and Y. Yue, “Towards deep radar perception for autonomous driving: Datasets, Methods, and Challenges,” 2022.

[34] M. Bijelic, T. Gruber and W. Ritter, “Benchmarking Image Sensors under Adverse Weather Conditions for Autonomous Driving,” *IEEE Intelligent Vehicles Symposium, Proceedings,* Vols. 2018-June, no. Iv, pp. 1773-1779, 2018.

[35] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang and A. Wallace, *RADIATE: A Radar Dataset for Automotive Perception in Bad Weather,* arXiv, 2020.

[36] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo and M. A. Kaafar, “The Impact of Adversary Weather Conditions on Autonomous Vehicles,” *IEEE Vehicular Technology Magazine,* 2019.

[37] Y. Zhang, A. Carballo, H. Yang and K. Takeda, “Perception and sensing for autonomous vehicles under adverse weather conditions: A survey,” *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 196, p. 146–177, February 2023.

[38] S. Hasirlioglu and A. Riener, “Challenges in Object Detection Under Rainy Weather Conditions: Second EAI International Conference, INTSYS 2018, Guimarães, Portugal, November 21–23, 2018, Proceedings,” 2019, pp. 53-65.

[39] J.-G. Wang, S. Chen, L.-B. Zhou, K. Wan and W.-Y. Yau, “Vehicle Detection and Width Estimation in Rain by Fusing Radar and Vision,” 2018.

[40] R. Yu, X. Li and T. Bi, “Modelling and validation of LiDAR noise distribution in fog and rain,” *Measurement,* vol. 229, p. 114472, April 2024.

[41] K. Wang, T. Zhou, X. Li and F. Ren, “Performance and Challenges of 3D Object Detection Methods in Complex Scenes for Autonomous Driving,” *IEEE Transactions on Intelligent Vehicles,* vol. 8, p. 1699–1716, February 2023.

[42] Y. H. T. D. Reddy Nagarjun and B. V. Arem, “Operational Design Domain Requirements for Improved Performance of Lane Assistance Systems: A Field Test Study in The Netherlands,” *IEEE Open Journal of Intelligent Transportation Systems,* vol. 1, p. 237–252, 2020.

[43] M. Kutila, P. Pyykönen, H. Holzhüter, M. Colomb and P. Duthon, “Automotive LiDAR performance verification in fog and rain,” 2018.

[44] M. Jokela, M. Kutila and P. Pyykönen, “Testing and validation of automotive point-cloud sensors in adverse weather conditions,” *Applied Sciences (Switzerland),* vol. 9, no. 11, 2019.

[45] M. Hadj-Bachir and P. De Souza, “LIDAR sensor simulation in adverse weather condition fordriving assistance development.,” *HAL,* 2019.