



## D3.3 Report on final AI algorithm development

Dissemination level	Public (PU)
Work package	WP3
Task:	T3.1, T3.2, T3.3, T3.4 and T3.5
Deliverable lead:	TUE
Version	V1.0
Submission date	17/07/2025
Due date	30/06/2025

## AUTHORS

Authors in alphabetical order		
Name	Organization	Email
Nerea Aranjuelo	VICOM	naranjuelo@vicomtech.org
Edin Arnautovic	TTTA	edin.arnautovic@tttech.com
Till Beemelmans	IKA	till.beemelmans@ika.rwth-aachen.de
Michiel Braat	TNO	michiel.braat@tno.nl
Maren Buermann	TNO	maren.buermann@tno.nl
Jean-Pierre Busch	IKA	jean-pierre.busch@ika.rwth-aachen.de
Paola Natalia Cañas	VICOM	pncanas@vicomtech.org
Alejandro Hernández	VICOM	ahernandeza@vicomtech.org
Aitor Iglesias	VICOM	aiglesias@vicomtech.org
Adwait Chandorkar	BUW	chandorkar@uni-wuppertal.de
Marc Facerias	IDIADA	marc.facierias@idiada.com
Raul Ferreira	CAF	raul.ferreira@conti-engineering.com
Alexandru Forrai	SIE-NL	alexandru.forrai@siemens.com
Mikel García	VICOM	mgarcia@vicomtech.org
Konstantinos Gkentsidis	SIE-BE	Konstantinos.gkentsidis@siemens.com
Guido Linden	IKA	guido.linden@ika.rwth-aachen.de
Ilma Okanovic	VIF	ilma.okanovic@v2c2.at
Jos den Ouden	TUE	j.h.v.d.ouden@tue.nl
Jan-Pieter Paardekooper	TNO	jan-pieter.paardekooper@tno.nl
Marc Perez	IDIADA	marc.perez@idiada.com
Bharat Shinde	Valeo	bharat.shinde@valeo.com
Mathieu Sarrazin	SIE-BE	Mathieu.sarrazin@siemens.com
Georg Stettinger	IFAG	georg.stettinger@infineon.com
Kristen van Strijp	MAPtm	Kristen.vanstrijp@maptm.nl
Sergi Vidal	IDIADA	sergi.vidal.bazan@idiada.com
Marijke van Weperen	TNO	marijke.vanweperen@tno.nl
Anton Wijbenga	MAPtm	anton.wijbenga@maptm.nl

## CONTROL SHEET

Version history			
Version	Date	Modified by	Summary of changes
0.1	07/03/2025	Jos den Ouden	Initial ToC
0.2	07/04/2025	Jos den Ouden, Paola Natlia Cañas Rodriguez, Michiel Braat, Alejandro Hernández, Guido Linden	First draft
0.3	12/05/2025	Jos den Ouden, Till Beemelmans., Marc Perzez, Marc Facerías Pelegrí, Michiel Braat	Reviewed and revised 2 <sup>nd</sup> draft input from partners
0.4	8/6/2025	Kristen van Strijp, Jos den Ouden	Final 2 <sup>nd</sup> draft input MAPtm, updated chapter 4
0.5	18/6/2025	All	Reviewed and revised 3 <sup>rd</sup> draft input from partners, ready for internal review
0.6	26/06/2025	Jos den Ouden, all	Received internal reviewers' comments and fixed partly
1.0	17/07/2025	Jos den Ouden	Removed contributors' names from titles (embedded in text now). Final version, ready for submission to ECAS Portal

Peer review		
	Reviewer name	Date
Reviewer 1	Pedro Brandimarte (VIC)	26/06/2025
Reviewer 2	Konstantinos Gkentsidis (SIE-BE)	24/06/2025



**Funded by  
the European Union**

**Project funded by**



Schweizerische Eidgenossenschaft  
Confédération suisse  
Confederazione Svizzera  
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs,  
Education and Research EAER  
**State Secretariat for Education,  
Research and Innovation SERI**

**Disclaimer**

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 CINEA. Neither the European Union nor the granting authority can be held responsible for them.

This work has received funding from the Swiss State Secretariat for Education, Research, and Innovation (SERI).

## TABLE OF CONTENTS

Authors .....	2
Control sheet .....	3
Table of COntents .....	4
1 Introduction .....	6
1.1 Athena concept and approach .....	6
1.2 Purpose of this deliverable .....	6
1.3 Structure of this deliverable .....	7
2 ML Development and life cycle management framework .....	9
3 Data and information fusion to reduce conflicting perception .....	10
3.1 Introduction .....	10
3.2 Explainable perception modules .....	11
3.3 Transformer-based Saliency Maps .....	13
3.4 Perception Model Robustness .....	17
3.5 Perception Model Optimization .....	21
3.6 Model Uncertainty .....	25
3.7 LiDAR-Radar-Camera Fusion .....	28
3.8 LiDAR-Camera Fusion .....	31
3.9 Explainable Object Level Fusion .....	31
3.10 Local Dynamic Maps .....	34
4 Transparent edge-case training using mixed real-synthetic data and safety-critical and trustworthy predictive models .....	35
4.1 Introduction .....	35
4.2 Collision risk prediction .....	36
4.3 Safety-critical and trustworthy prediction models .....	39
4.4 Context understanding through VLM .....	40
4.5 Domain adaptation strategy for training with real and synthetic data .....	42
4.6 AI algorithm for application in UC2.2 .....	44
5 Explainable and robust decision making (manoeuvre and trajectory) .....	60
5.1 Introduction .....	60
5.2 Hybrid AI motion planning .....	61
5.3 Generating trajectories for critical driving scenarios .....	65
5.4 Situation awareness: Competence, context, and risk .....	65
5.5 Network level impact of introducing AVs .....	69
6 Conclusions .....	77

7	Bibliography.....	78
---	-------------------	----

## 1 INTRODUCTION

### 1.1 Althena concept and approach

Connected, Cooperative and Automated Mobility (CCAM) solutions have emerged thanks to novel Artificial Intelligence (AI) which can be trained with huge amounts of data to produce driving functions with better-than-human performance under certain conditions. The race on AI continues to build hardware (HW) and software (SW) frameworks to manage and process even larger real and synthetic datasets to train increasingly accurate AI models. However, AI remains largely unexplored with respect to explainability (interpretability of model functioning), privacy preservation (exposure of sensitive data), ethics (bias and wanted/unwanted behaviour), and accountability (responsibilities of AI outputs). These features will establish the basis of trustworthy AI as a novel paradigm to fully understand and trust AI in operation, while using it at its full capabilities for the benefit of society. Althena will contribute to build Explainable AI (XAI) in CCAM development and testing frameworks, researching three main AI pillars: data (real/synthetic data management), models (data fusion, hybrid AI approaches), and testing (physical/virtual X-in-the-loop (XiL) set-ups with scalable Machine Learning Operations (MLOps)). A human-centric methodology will be created to derive trustworthy AI dimensions from user identified group needs in CCAM applications. Althena will innovate by proposing a set of Key Performance Indicators (KPI) on XAI and an analysis to explore trade-offs between these dimensions. Demonstrators will show the Althena methodology in four critical use cases: perception (what does the AI perceive, and why), situational awareness (what is the AI understanding about the current driving environment, including the driver state), decision (why a certain decision is taken), and traffic management (how transport-level applications interoperate with AI-enabled systems operating at vehicle level). Created data and tools will be made available via European data sharing initiatives (OpenData and OpenTools) to foster research on trustworthy AI for CCAM.

### 1.2 Purpose of this deliverable

This deliverable D3.3 is the third and final output of WP3 (model development) and is related to all tasks of WP3. This deliverable focuses on reporting the final designs of the AI algorithms that have been developed within AITHENA, according to the AITHENA pipeline. This report covers the final AI algorithm developments including their final feature validation according to the specified requirements and specifications.

Additionally, all strength and weaknesses of the specific AI-driven approaches are outlined

This deliverable also incorporates latest developments using the AITHENA ML Framework that was already published in D3.1. As mentioned in D3.1, the results of Task 3.1 are also embedded in D3.2 and D3.3.

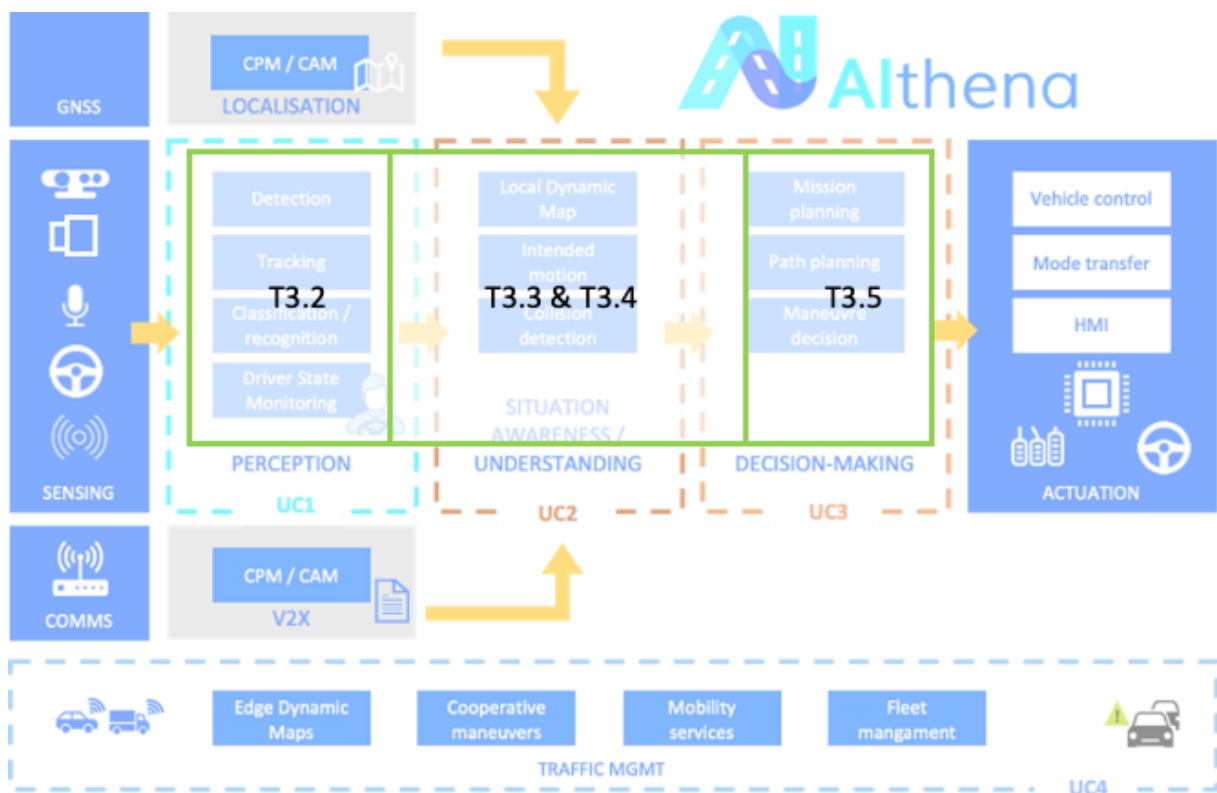
This deliverable provides a first iteration towards achieving the following general objectives of Althena:

- XAI - GO-2: XAI models – research and development of explainable AI models,
- DATA - GO-5: Development Framework (“DevOps” like) to ensure data and model lifecycle tracking and management,
- DATA - GO-6: Digital Twin for data generation,

- TEST - GO-7: Testing and validation procedures – methodology for extending HEADSTART methodology to include AI based functions and systems,
- CCAM - SO-1: Trustworthy and Robust Perception systems,
- CCAM - SO-2: Human Understandable Situation Awareness System including driver state,
- CCAM – SO-3: Explainable Driving Decision methods.

### 1.3 Structure of this deliverable

According to the grant agreement, tasks 3.2, 3.3, 3.4 and 3.5 focus on the different parts of the perception, situation awareness / understanding and decision-making pipeline as present in the following architecture (Figure 1):



**Figure 1: AITHENA architecture, with relationship of tasks T3.2, T3.3, T3.4 and T3.5 integrated.**

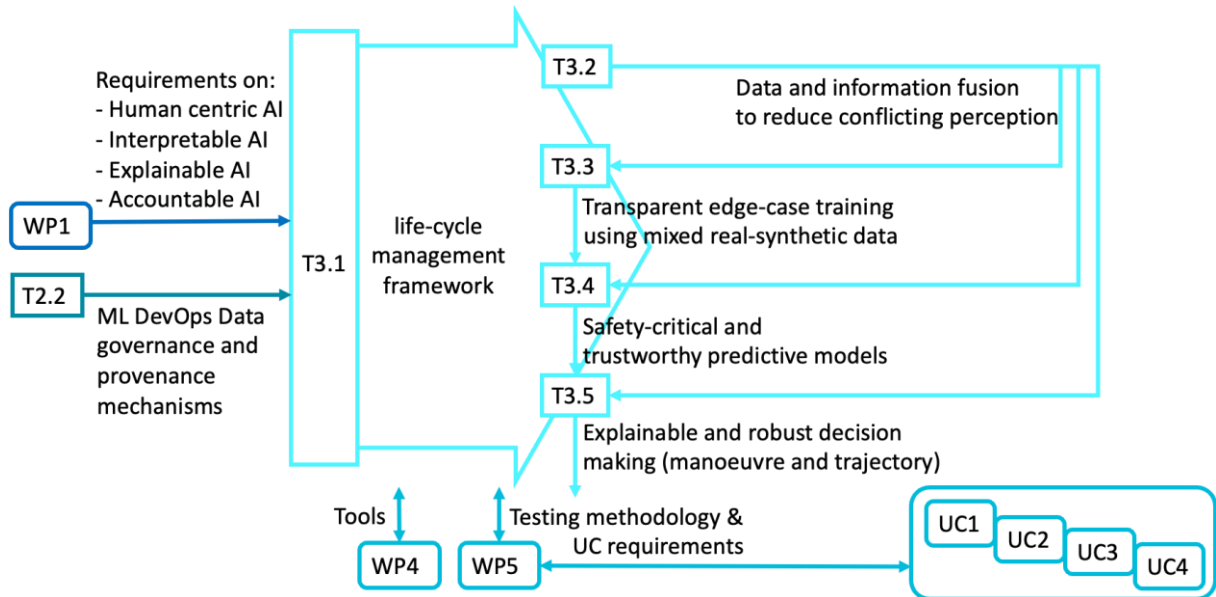
All the AI algorithms that have been developed in Aithena are developed in these 4 tasks. Additionally, task 3.1 provided the machine learning life-cycle management framework. Figure 2 provides an overview of how WP3 tasks fit together, how WP1 and WP2 provide input to WP3, and how WP4 and WP5 are interacting (both ways) with WP3.

WP1 and T2.2 provided input with the requirements and ML DevOps data governance structure to task 3.1 which provided an overview of a ML life-cycle management framework.

In parallel the four tasks T3.2 – T3.5 focus on the individual parts as described in Figure 1 already above.

In WP4 the tools for the development and testing of the AI algorithms are developed and in WP5 the testing methodology and UC requirements are defined.

Specifically, D5.3 has a direct link to this deliverable, as evaluation results of several AI algorithms described in this deliverable are being described in full detail there to link them more concretely to use cases. D3.3 continued to focus therefore (similar to D3.2 [1]) more on the final description and methodologies used for developing these AI algorithms, with only a extensive algorithmic result.



**Figure 2: structure of WP3 with respect to other work packages and use cases**

As development went, some parts of research have been stopped or did not generate any other results than those already described in D3.2. To make this clear, the same structure of D3.2 has been replicated here and where no new results were generated, this is clearly stated. In some cases, research has been shifted to new topics and for those additional sections have been created.

This deliverable is structured as follows:

Chapter 2 provides a short update on the ML Framework (T3.1) that was previously introduced in D3.1. Since this task only had one deliverable while the framework remained in continuous development, we provide an update in this deliverable.

Chapter 3 focuses on the data and information fusion to reduce conflicting perception (linked to work in T3.2).

Chapter 4 focuses on the transparent edge-case training using mixed real-synthetic data (linked to the work in T3.3) and on safety-critical and trustworthy predictive models (linked to the work in T3.4).

Chapter 5 focuses on explainable and robust decision making (manoeuvre and trajectory) (linked to the work in T3.5)

Chapter 6 provides a conclusion.

## 2 ML DEVELOPMENT AND LIFE CYCLE MANAGEMENT FRAMEWORK

In the previous deliverable 3.1. Life-cycle management framework for ML models [2], a description of the ML (machine learning) development cycle has been presented. A special contribution was the definition of a model and MLOps (Machine Learning Operations) card. These tools not only promote transparency but also encourage standardization for reporting relevant information to different potential users.

Our defined model card proposes a series of sections inspired by Mitchell et al.'s original model card concept, with the addition and reordering of some sections to include ML developments within the CCAM field. In alignment with Athena's human-centric design, the model card is complemented by a checklist. This assists developers or creators in providing pertinent details about their model or system, ensuring that comprehensive and relevant information is shared. This model card methodology has been implemented and tested in AITHENA by IKA and VIF. These documents can be found on the following links: [IKA](#), [VIF](#).

There has not been any updates in the Model Card definition template since the ones reported in D3.2. An overview of strengths and weaknesses of this framework based on the human centric AI principles as defined in D1.1 [3], [4] is provided in Table 1.

**Table 1: Explainable perception modules - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	Details about how the testing process was performed (representativeness of the test data) should be included in the model card. This can inform about the bias present in the model	These model cards are expected to be completed by model's authors and developers. The content can be somehow technical and less clear in use of wording towards the public.
<b>Transparency</b>	One of the main purposes of these cards is to increase transparency. The communication of model's limitations, risks and other relevant details is done through this methodology.ad	Although this methodology encourages developers to disclose various relevant details and information about their model, Intellectual property protection policies can make the information disclosure limited
<b>Accountability</b>	In the model card presented, contact details and information about the authors and developers are requested. Also, methods to the user to ask questions, express their concerns and report incidents.	N/A
<b>Privacy</b>	Information about how the authors ensure that the privacy of the user is	N/A

preserved is also included in this methodology.

### 3 DATA AND INFORMATION FUSION TO REDUCE CONFLICTING PERCEPTION

#### 3.1 Introduction

This chapter details the completed development within Task 3.2, which successfully built trustworthy and interpretable perception systems for autonomous vehicles. The achievement of these goals was realized through a multifaceted development approach incorporating Explainable Artificial Intelligence (XAI), robust model design, and advanced sensor modality fusion.

Trust in autonomous vehicle perception systems was established through the successful implementation of two key aspects: interpretability and robustness. Interpretability allows humans to understand how the developed system arrives at its decisions, thereby fostering trust and acceptance. Robustness ensures reliable object detection across diverse environmental conditions, significantly enhancing the system's credibility.

Within Task 3.2, various XAI techniques were successfully implemented to render the perception models interpretable. This included integrating transparent layers within the model architecture to reveal the rationale behind its decisions, resulting in the development of Explainable Perception Modules. Furthermore, efficient Model Uncertainty Quantification techniques were implemented to determine model limitations during runtime and highlight rare or unseen objects.

Strategies to enhance multi-modal model robustness were developed and implemented. This involved establishing a new dataset and benchmark specifically designed for evaluating the performance of state-of-the-art perception models under challenging conditions. Using this benchmark, existing state-of-the-art models were systematically tested and evaluated for their robustness against various types of corruptions.

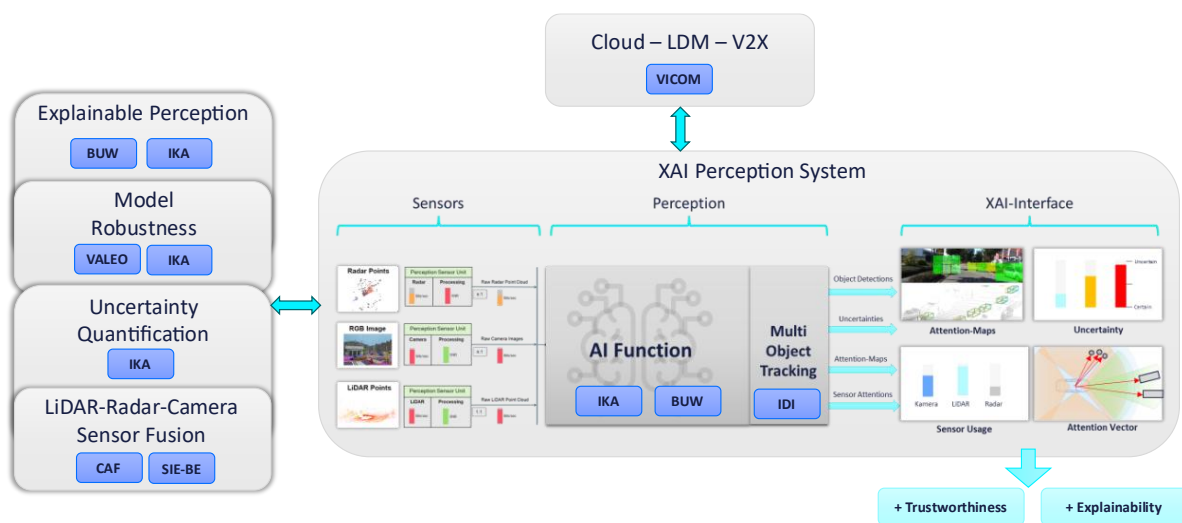


Figure 3: Conceptual overview of WP3.2 with the involved partners.

The development within Task 3.2 successfully contributed to improving Local Dynamic Maps (LDMs). These real-time databases fuse static and dynamic environmental information from various sources, now effectively integrating the outputs from the developed perception system. By providing a comprehensive and up-to-date representation of the environment derived in part from the perception system's data, the enhanced LDMs play a crucial role in supporting explainable perception.

Ultimately, the research and developments within Task 3.2 contributed to the UC1 demonstrator by ensuring the trustworthiness, interpretability, and robustness of the perception system, paving the way for the safe and reliable operation of autonomous vehicles. A conceptual overview of this WP including the involved partners is shown in Figure 3.

## **3.2 Explainable perception modules**

The goal is to make all perception modules which are developed in this working package explainable during inference. While some architectures offer inherent mechanisms that can be exploited for explanation, like self-attention layers in transformers, other architectures may not. BUW has developed and tested a model-agnostic explainability method usable for all involved partners. The following necessary criteria were identified in collaboration with the involved partners:

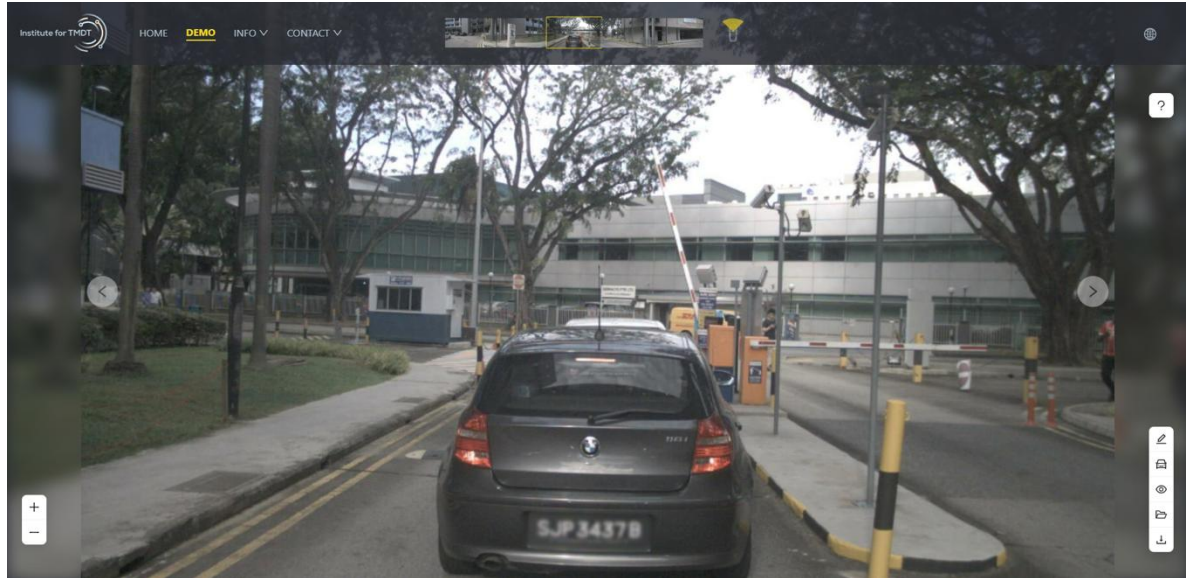
- Analysis of features in the input that the model focuses on for object classification.
- Identification of learned biases towards certain features.
- Explanations for specific misclassifications.

### **3.2.1 Methodology**

BUW has identified Detector Randomized Input Sampling for Explanation [5] (D-RISE) as a suitable state-of-the-art baseline method. While D-RISE has been shown to provide robust explanations, it is limited to single image inputs and requires too many inferences to be real-time capable in its current form. BUW's first research contribution is extending D-RISE to accommodate multi-sensor inputs and enable offline detection. To this end, BUW has implemented the XAI method with the nuScenes dataset, as most of the latest state-of-the-art models benchmark their results on this dataset, and worked through the state-of-the-art of multi-image object detection models for autonomous driving. Our results of extending the scope of D-RISE to multi-sensor inputs on the PETR algorithm [6] demonstrate promising results to provide comprehensive explanations on multi-sensor fusion models that are in line with real-world scenarios and expert reasoning.

### **3.2.2 Results**

To demonstrate the functioning of the XAI method, we have developed an online demonstrator which visualizes the explanations for the predictions of the PETR model. Users can freely draw their own objects and/or mask certain parts of the image to see how the prediction of the model changes. In BUW's opinion, the demonstrator gives a better understanding of how the complex deep learning models work, and which regions of the object are important in decision-making. As the current BUW infrastructure doesn't include a GPU for the online demonstrator, the process is slower. BUW has raised a request with the concerned IT team for the required resources and is hopeful that the demonstrator can be implemented on a GPU on the public event planned in October 2025.



**Figure 4: Online XAI Demonstrator developed by BUW**

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 2.

**Table 2: Explainable perception modules - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	Fairly evaluates the strength/limitations of any 3d object detector based as the approach is model agnostic. Moreover, our XAI demonstrator also enables the users to draw/paint additional objects and then evaluated if they are predicted correctly.	Currently limited to only camera-based 3d object detectors.
<b>Transparency</b>	The method generates saliency maps which visualize which pixels are important for decision making.	Though saliency maps is a standard approach to generate explanations, it fails to visualize the complex decision-making approach of the deep-learning models.
<b>Accountability</b>	Saliency maps provide visual evidence of which inputs influenced a decision, which can support accountability processes	Current saliency maps generation process is slow and gets influenced by the background which may or may not be relevant towards prediction.
<b>Privacy</b>	Since the method is model agnostic, meaning unaware of the model parameters, it ensures that the privacy is maintained.	-

## 3.3 Transformer-based Saliency Maps

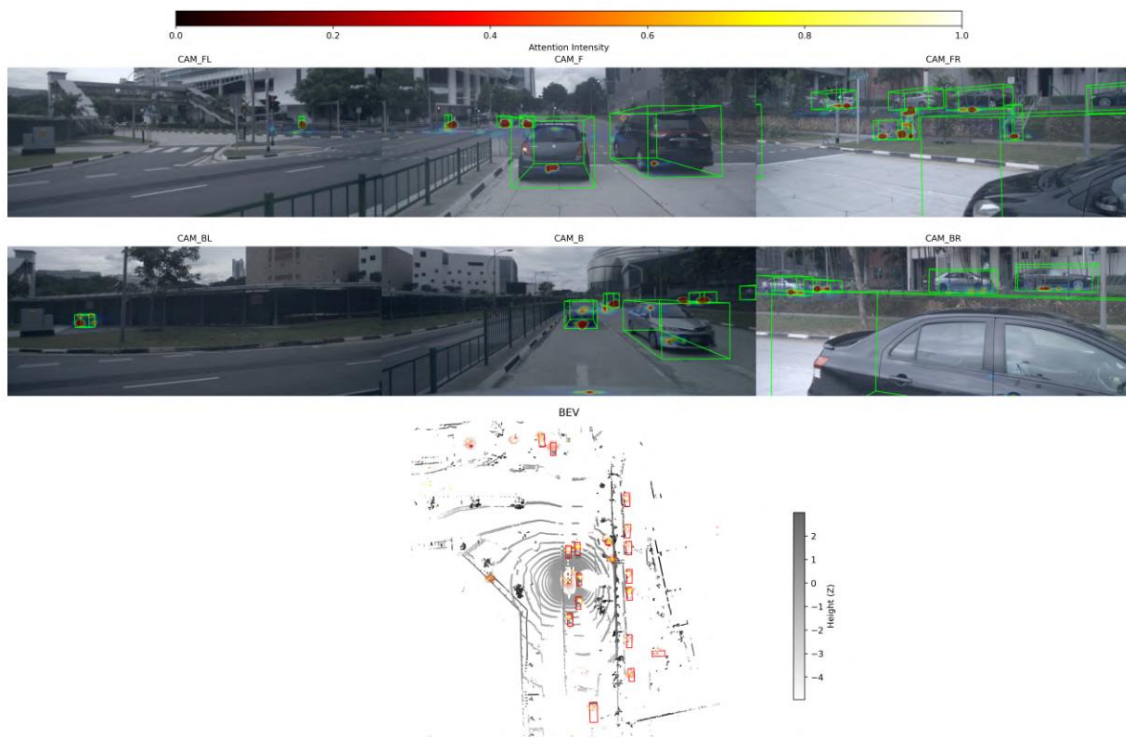
### 3.3.1 Introduction

Vision Transformers (ViTs) have been successfully implemented and integrated on a wide range of scene understanding tasks critical for automated driving. The core principle leveraged is the transformers' attention mechanism, which was successfully adapted to capture a global understanding of the scene. This adaptation enables more accurate detections while effectively eliminating the need for traditional handcrafted postprocessing or object fusion steps. This end-to-end approach, while offering significant advantages, presented a challenge in terms of explainability. This challenge was particularly crucial considering the deployment of these advanced ViT-based systems in safety-critical autonomous driving applications. In such scenarios, it is essential for authorities, developers, and users to have a clear and trustworthy understanding of the model's reasoning behind its predictions, especially when integrating multiple sensor modalities like camera and LiDAR.

### 3.3.2 Methodology

Previously, raw attention maps were successfully used for camera-based 3D object detection models [7]. Addressing the explainability challenge in a **multi-modal** context, IKA successfully developed and implemented a saliency map generation approach specifically for a **multi-modal transformer model** that fuses **camera** and **LiDAR** data. This approach aimed to enhance the understanding of the model's behaviour and provide valuable insights into which regions of the input data from both camera images and LiDAR point clouds were most influential in determining object detection predictions. Saliency maps, as successfully generated by this method, are visualizations that effectively highlight the regions in the input modalities that the model considers most influential for its prediction. These visualizations are presented as heatmaps overlaid on the original data, where brighter areas indicate elements, the model deemed most important for its decision.

The successfully implemented method by IKA generates these *multi-modal saliency maps* efficiently during the runtime of the model, demonstrating superior performance compared to computationally expensive gradient-based methods. The effectiveness and reliability of this developed approach were rigorously validated through comprehensive perturbation tests applied to both camera and LiDAR inputs. Figure 4 provides a conceptual overview of this developed methodology.



**Figure 4: IKA investigated how multi-modal 3D object detector can be extended to generate saliency maps as visual explanations for a multi-modal transformer-based 3D object detector can be efficiently generated.**

In this approach, the model's attention layers are used to produce saliency maps for the interactions in the cross- and self-attention. In the following, different gradient-free and gradient-based computation and propagation rules are presented.

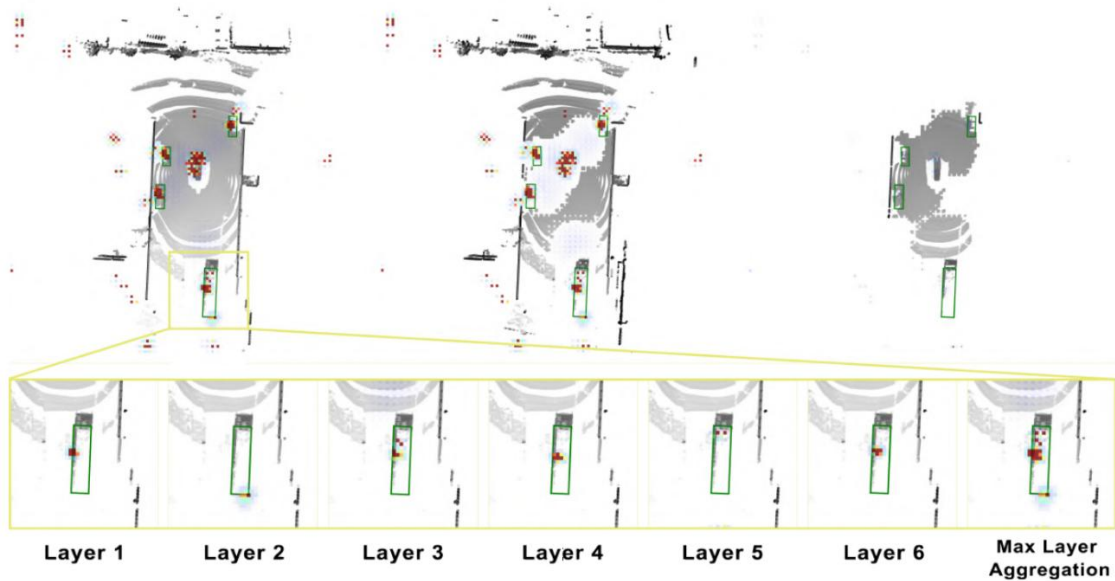
#### Gradient-Based Methods

- Grad-CAM

#### Raw-Attention Methods

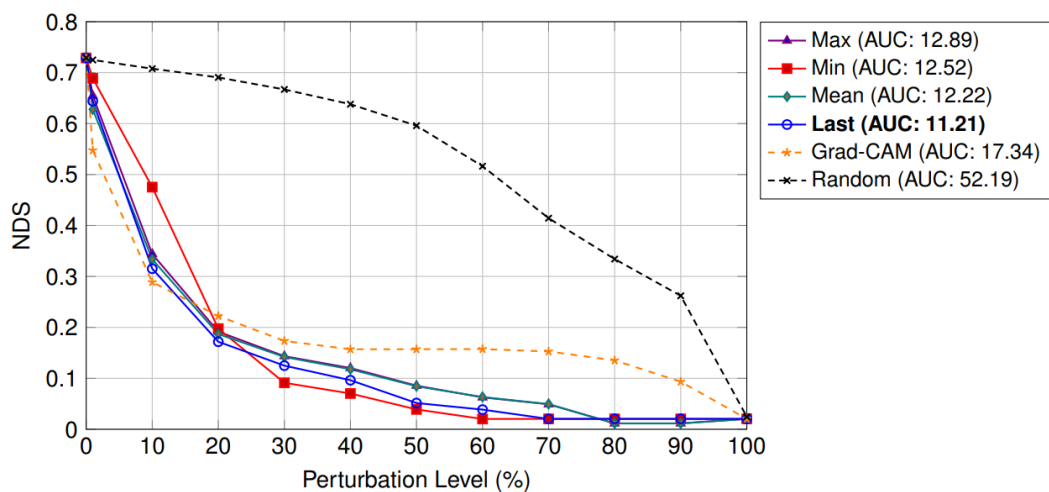
- Raw Attention Last Layer
- Raw Attention Mean-Layer Fusion
- Raw Attention Max-Layer Fusion

IKA performed extensive experiments on the nuScenes dataset. First, a qualitative visual exploration of the attention maps in the point cloud domain was conducted, as shown in Figure 5.



**Figure 5: Bottom: Saliency maps for a LiDAR point cloud generated for several layers in the transformer architecture. Top: Point cloud perturbation is shown with increasing perturbation level from left to right. Points from the point cloud are removed from the original point cloud during perturbation.**

Second, a quantitative evaluation tested the quality produced by each method using positive and negative perturbation tests was conducted, as shown in Figure 6. For the positive perturbation test, pixels and LiDAR points with high attention scores were progressively removed or masked from the input. A step decrease in the final detection accuracy with increasing perturbation level in this test indicates good explainability of the XAI method, as it suggests the method successfully identified and removed the most important input features. Conversely, for the negative perturbation test, pixels and LiDAR points with the lowest attention scores were masked. In both scenarios, at each perturbation level (e.g., 10%), all top-k or bottom-k pixels were removed during evaluation, and the NuScenes Detection Score was subsequently evaluated using the perturbed input. A lower score (or AUC) following perturbation signifies that the masked regions were indeed more relevant for object detection, thereby confirming the efficacy of the XAI method.



**Figure 6: Positive perturbation test evaluated on the nuScenes validation set. Smaller AUC is better.**

The model's end-to-end implementation successfully generates a single detection for objects visible across multiple camera views or detected by LiDAR. For objects appearing in multiple camera views, the developed method successfully visualizes the attention distribution across the overlapping image regions, as illustrated in Figure 7. This implemented approach also successfully increases the traceability of the detector network by estimating the contribution of each camera view (and the influence of LiDAR) to a detection, as indicated by the blue bars in Figure 7.



**Figure 7: Raw cross-attention examples for objects (green OBB) that lie in the overlapping FOV of two cameras. In each example, a single query is used to generate the saliency maps for all camera images. Attention can be observed on the object on both overlapping images.**

### 3.3.3 Results

The developed multi-modal saliency approach, based on raw cross-attention, proved significantly more efficient than gradient-based methods. Perturbation tests successfully demonstrated its effectiveness and reliability across camera and LiDAR inputs. Visual analysis of attention maps showed the transformer's ability to alternate focus across layers and modalities to distinguish classes and determine properties. Aggregating attention across layers was crucial for comprehensive explanations. These insights successfully pave the way for the developed explainable 3D object detector within WP3 and its integration into the UC1 demonstrator.

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 3.

**Table 3: Transformer based saliency maps - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	-	Saliency maps visualize model focus but do not guarantee the model is free from bias or that it treats all inputs/groups fairly.
<b>Transparency</b>	Successfully generates multi-modal saliency maps to highlight influential input regions, enhancing understanding and increasing traceability of the detector network.	Saliency maps provide a visual explanation but may not fully convey the complex logical reasoning or interaction of features within the model.
<b>Accountability</b>	Saliency maps provide visual evidence of which inputs influenced a decision, which can support accountability processes	The saliency map method itself is not an accountability mechanism; it is a tool that aids human understanding for potential accountability.
<b>Privacy</b>	N/A	N/A

### 3.4 Perception Model Robustness

Autonomous Vehicles (AVs) must comprehend their surrounding environment—vehicles, pedestrians, cyclists, and their respective postures—to further estimate the speed or future trajectories of these moving objects and plan their own motions accordingly. While advancements have been made in the domain of autonomous driving perception, most testing and training of perception models is conducted under optimal weather and road conditions with clear visibility. Urban noise, weather and traffic conditions significantly impact the safety and operability of AVs. For autonomous vehicles to gain widespread acceptance and trust, they must demonstrate robustness, reliability and accuracy under adverse weather and road conditions. Within WP3, model robustness for LiDAR-Camera fusion (see 3.4.1) and weight fusion on the robustness for the CLIP [8] model (see 3.4.4) have been explored.

#### 3.4.1 Robustness LiDAR-Camera Fusion – Introduction

Regarding Multi-Modal sensor setups with multiple cameras and LiDARs, additional issues like miscalibration during the vehicle's motion, and sensor misalignments in terms of varying frequencies or latencies often lead to deviations between sensor modalities. These problems are of particular interest when multi-modal detection methods are employed. Their effectiveness and robustness largely depend on how and where information is fused within the model.

#### 3.4.2 Robustness LiDAR-Camera Fusion - Methodology

Building on these considerations, a consistent and open-source evaluation framework for assessing the robustness of multi-modal detection algorithms has been developed by IKA. The evaluation dataset and framework - named MultiCorrupt [9] - consist of 10 multi-modal sensor corruptions and is open sourced at <https://github.com/ika-rwth-aachen/MultiCorrupt>, allowing researchers to easily test and benchmark their own models.

### 3.4.3 Robustness LiDAR-Camera Fusion - Results

As a completed task within the project, we extended our previous evaluation by **additional six state-of-the-art multi-modal detector approaches** on the MultiCorrupt benchmark, concluding our analysis of their robustness against various corruptions based on their fusion strategies. We added

- Two variants of the MEFoformer Architecture
- One variant of the MoME Architecture
- Three variants of UniBEV Architecture

to our evaluation benchmark, as summarized in Figure 8.

Our findings revealed that **independent modality** handling, **ensemble learning** approaches and **masked-modal training** enhanced robustness, while modality-dependent query initialization and early deep feature coupling diminished it. This work has successfully provided critical insights into multi-modal 3D object detection robustness, directly contributing to the development of a trustworthy detector for WP3 and UC1.

Model	Clean	mRRA	Beams Red.	Brightness	Darkness	Fog	Missing Cam.	Motion Blur	Points Red.	Snow	Spatial Mis.	Temporal Mis.
MEFormer	0,739	7,391	16,759	0,878	1,556	8,586	-0,168	-0,307	8,865	11,382	16,689	9,668
CMT	0,729	7,161	18,642	-1,138	-0,096	9,398	2,041	-0,841	8,213	9,887	17,053	8,448
MoME	0,735	6,837	15,061	0,400	0,694	7,891	1,634	-0,685	8,421	10,852	14,975	9,128
MEFormer w/o PME	0,737	6,740	15,172	0,714	1,522	8,176	-0,791	-0,679	8,384	10,749	14,968	9,184
Sparsefusion	0,732	3,033	4,264	3,179	1,821	4,429	0,297	0,280	3,242	1,887	3,699	7,228
IS-Fusion	0,737	2,708	3,684	2,291	1,267	3,890	0,920	3,994	1,691	-2,351	4,513	7,177
BEVfusion	0,714	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
TransFusion	0,708	-1,718	-7,210	1,799	1,146	-0,552	0,340	-5,412	-3,296	-4,220	-3,626	3,850
UniBEVavg	0,684	-2,154	7,617	-3,758	-4,595	-1,228	-5,170	-11,777	-0,144	-6,909	2,812	1,617
UniBEVcat	0,678	-2,653	6,534	-4,303	-5,279	-0,199	-5,438	-12,505	-0,979	-6,596	1,436	0,799
UniBEVcnw	0,685	-2,893	5,030	-3,729	-4,383	-1,104	-5,749	-13,119	-0,428	-8,055	1,582	1,025
DeepInteraction	0,691	-7,221	-6,361	-3,150	-7,215	-25,03	-16,38	-7,077	-2,188	-5,149	0,212	0,145

**Figure 8: Results of the Robustness Evaluation of state-of-the art 3D Object Detection models for all corruptions and severity levels. Robustness Ability for different severity levels computed using NDS score.**

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 4.

**Table 4: Robustness LiDAR-Camera Fusion - Strength and Weaknesses**

	Strengths	Weaknesses
Fairness	-	No separate evaluation of performance for diverse humans (e.g., pedestrians of varied age, appearance) possible.
Transparency	Open-source evaluation framework allows external auditing of model robustness to specific corruptions.	No discussion on the interpretability or explainability of the perception models themselves ("why" a detection was made or missed).

<b>Accountability</b>	Improved robustness aims to enhance safety and reliability.	No mechanisms for assigning responsibility or accountability in case of perception system failure.
<b>Privacy</b>	No additional sensitive data was generated, as the corruptions are simulated into an existing public dataset.	-

### 3.4.4 Weight Fusion – Introduction

Weight fusion is an emerging mechanism that aims to replicate the generalization capability of deep ensembles without the disadvantage of increased runtime requirements. This work focuses on investigating the concrete impact of weight fusion on the foundation model CLIP [8], which combines the text and image modals.

Traffic sign classification is an elementary component of safe autonomous driving. The vast number of possible traffic signs, which differ in appearance between national borders, represent a major challenge. A huge amount of training data is needed to classify each sign accurately. CLIP has shown that it has an outstanding robustness capability in zero-shot cases. However, a pure zero-shot application of the basic CLIP model is insufficient. A fine-tuned CLIP model is necessary; however, it reduces the generalization capability of the model. Hence, a weight fusion approach based on the greedy procedure of model soups is tested to investigate the regeneration of generalization ability of CLIP.

### 3.4.5 Weight Fusion - Methodology

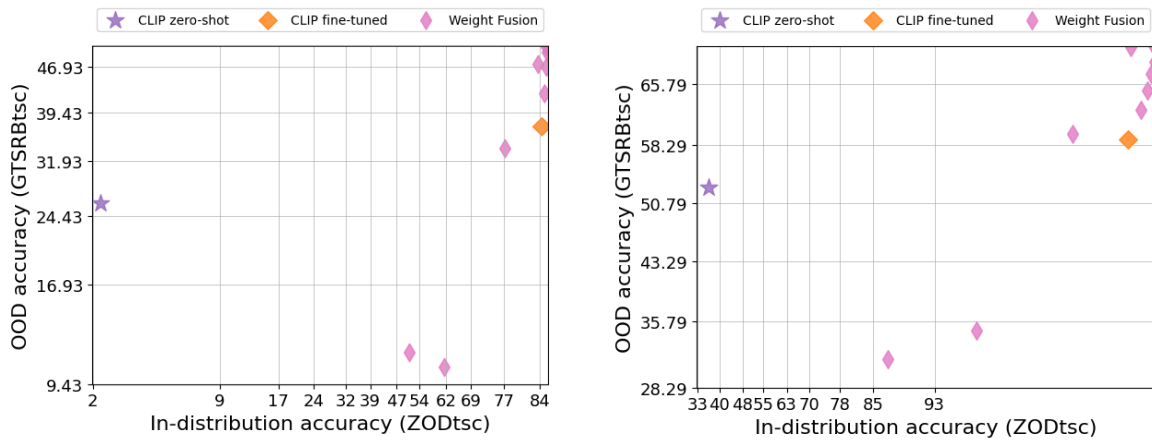
Based on the traffic sign classification, a variety of experiments were conducted to show the influence of weight fusion. To measure the prediction performance of CLIP, the top 1 and top 5 error rate were used as KPIs. To test the zero-shot performance, CLIP was initialized with the weights generated on 400 million image-text pairs during training. The three datasets GTSRB [10], Mapillary [11] and ZOD [12] were used to fine-tune CLIP. GTSRB is a dataset specifically created for benchmarking traffic sign recognition algorithms. It contains images of traffic signs commonly encountered on roadways in Germany. This dataset is the smallest of the three datasets and contains about 39k training data and 12.6k validation data. Mapillary is an extensive image and object recognition dataset that contains various scenes captured from road images. After extracting all relevant road signs and combining cross-national road signs with the same meaning, the dataset results in 236 classes. There are 180 sign instances available for training and 26 k for validation. The Zurich Object Detection (ZOD) dataset was developed for object detection tasks and contains images that were taken in urban environments and can be used for the classification of traffic signs. Approx. 800k training data and 83k validation data are available, in which 106 classes have been labelled.

CLIP was fine-tuned for 10 epochs. For the weight fusion approach, the model from the final epoch (epoch 10) was set as the baseline. This model was then fused with the previous epoch (epoch 9) using simple weight averaging with a fixed factor ( $\alpha = 0.5$ ). The resulting fused model was evaluated on a validation set using Top 1 and Top 5 error rates. If the new model outperformed the baseline, the fused weights were retained. This process was repeated, fusing the retained model with earlier checkpoints, each time using the same factor. Following the greedy model soup strategy, the fusion was retained only if it improved the performance. This was repeated until epoch 1. This approach

aggregates the most beneficial representation across the training epochs. It encourages the generalizable features learned in the initial epochs, while mitigating overfitted representations.

### 3.4.6 Weight Fusion - Results

Figure 11 shows the results after fine-tuning on ZOD (in-distribution): Weight fusion was found to have a significant influence on the robustness capability. If only the in-distribution performance is considered, weight fusion can also lead to an improvement in all cases.



**Figure 9: Left: Top 1 error rate; Right: Top 5 error rate**

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 5.

**Table 5: Weight Fusion - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	Improves generalization ability of the model which ensures consistent performance across all datasets and classes.	Might amplify biases if present in any dataset.
<b>Transparency</b>	Clear and consistent fusion strategy applied across all datasets.	Complex fusion strategy could lead to reduced interpretability.
<b>Accountability</b>	Improves model's robustness and enables better performance across diverse conditions, which contributes to safety and reliability of the system.	Difficult to attribute a specific result to a particular model or dataset.
<b>Privacy</b>	No sensitive data shared.	N/A

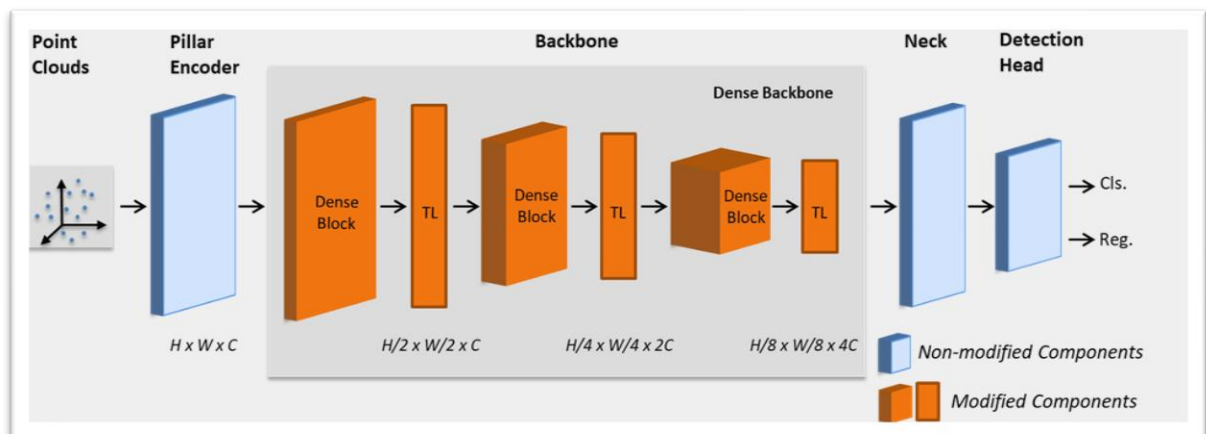
## 3.5 Perception Model Optimization

### 3.5.1 Introduction

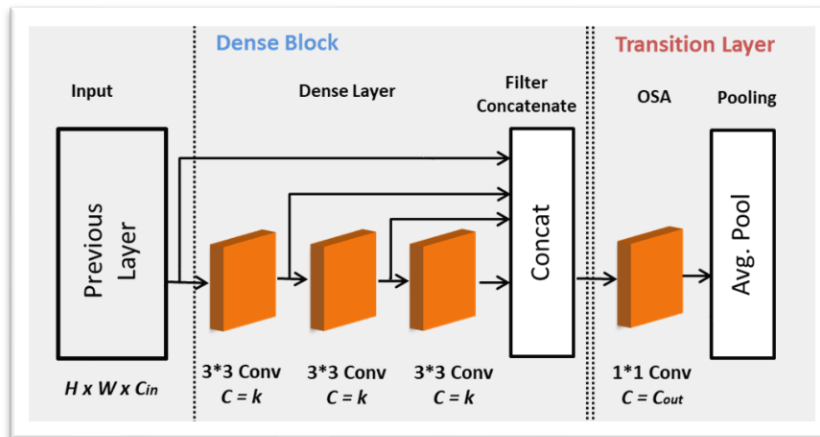
One of the key challenges for autonomous mobility is the computing power of the server in the vehicle. There could be multiple AI models running on the server such as object detection, path planning, manoeuvring etc., and most of them require extensive computing power. Unfortunately, there are limitations on the computing resources available in the vehicle as it is infeasible to install a high-end server owing to space and cost constraints. One approach to solving this problem would be to transfer the raw data to a cloud server, run the AI models, and transfer the results back to the vehicle. However, to ensure this, the vehicle must always drive in a secure, high-speed, and reliable network.

### 3.5.2 Methodology

To overcome these challenges, BUW proposed model compression as an alternative approach. Traditionally, this has been achieved either using quantization (typically 16-bit) or through model-pruning. BUW's approach uses a modification of the backbone network which results in model compression without significantly impacting the performance compared to the original model, as shown in Figure 10. Currently, BUW has implemented this approach only for the object detection task, but in the future, it can also be extended to tasks such as segmentation, tracking etc.



**Figure 10: Our DenseBackbone adopted in a 3D object detector. In the customization of the backbone network, we draw insights from existing models like PeleeNet [13] and VovNet [14], which propose state-of-the-art compression methodologies specifically designed for image data. Our approach combines the best of two models and extends the methodology for a 3D point cloud-based object detection.**



**Figure 11: Network architecture of our DenseBackbone. A *Dense Block* consists of multiple cascaded dense  $3 \times 3$  Conv layers. The *transition layer* aggregates the concatenated features from the Dense Block with a pointwise Conv layer followed by Pooling. The output channels of  $3 \times 3$  Conv are controlled by a hyperparameter *growth\_rate*, denoted by  $k$ .**

### 3.5.2.1 Design Rationale

Our objective is to minimize model parameters and computational cost by strategically reducing inter-layer connections. This can typically be achieved by increasing the stride or reducing the output channels in convolutional layers. However, higher strides tend to decrease spatial resolution and limit receptive field overlap, thereby impeding the capture of fine-grained features. Similarly, reducing the number of channels may hinder the model's ability to represent details if each channel fails to capture essential feature distinctions. This motivated our approach: first, to reduce the number of output channels to lower computational cost; and second, to mitigate the resulting loss in representational capacity by maximizing feature reuse across layers. This enables the model to effectively capture features at multiple spatial scales, effectively rethinking the conventional design of the backbone.

### 3.5.2.2 Design

**Dense Block:** The building block of Dense Backbone is the dense layer. Inspired by VoVNet [14], we employ a series of feed-forward Conv layers followed by concatenation at the end as shown in Figure 11. The concatenation of features enables simultaneous feature learning for objects with varying aspect ratios. **Transition Layer:** The transition layer is an intermediate layer between subsequent *dense blocks*. To avoid reducing the receptive field of the concatenated features we aggregate these features with a point-wise convolution layer. Finally, we incorporate an average pooling layer with a stride of 2. Our extensive studies prove that *pooling* layer offers better detection accuracy over strided convolutions. **Growth Rate:** The *growth\_rate* ( $k$ ) determines the number of learnable parameters in each dense layer. Fewer channels in the  $3 \times 3$  kernels certainly help in reducing network weight, but this can also impact feature learning. To overcome this, we modify  $k$  for each dense block to ensure there is no large disparity in concatenated channels and output channels  $C_{out}$ . We initialize  $k$  to 32 in the first dense block and progressively double it in each subsequent dense block. This strategy allocates more learnable parameters to deeper layers, enhancing their capacity to extract higher-level semantic features.

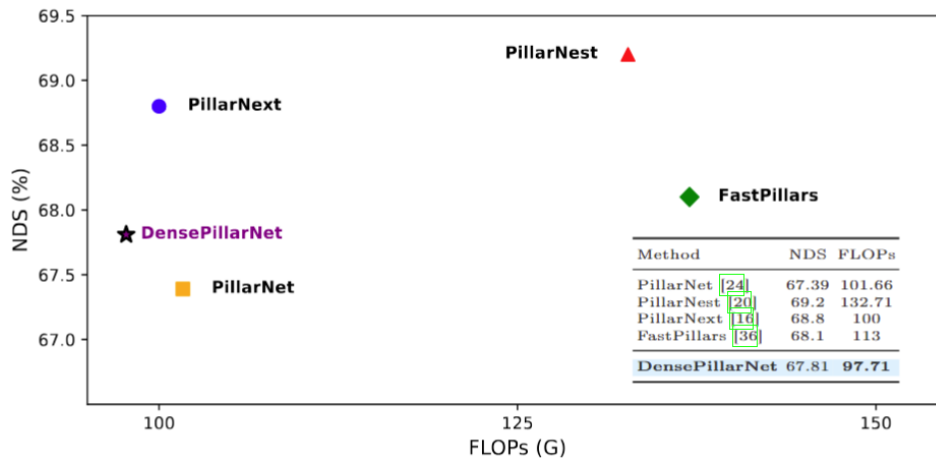
### 3.5.3 Results

#### 3.5.3.1 Selecting the baseline

To rigorously evaluate the plug-and-play capability and generalizability of our proposed *Dense Backbone*, we integrate it into three representative and widely adopted 3D object detection frameworks. First, we select PointPillars [15] as the base for experiments on the KITTI dataset given its simplicity, wide adoption, and strong performance. Second, we adopt CenterPoint [16] as the base model for the nuScenes dataset. CenterPoint’s detection head, *CenterHead*, has become the de facto standard for recent LiDAR-based detection pipelines due to its versatility and superior performance. Finally, we adapt our Dense Backbone into PillarNet [17]. PillarNet is a strong baseline for pillar-based detectors and is also used as a benchmark in recent works. Our adapted models are referred as DensePointPillars, DenseCenterPoint and DensePillarNet respectively.

#### 3.5.3.2 Results on State-of-the-art models

When evaluated on the nuScenes dataset, our DensePillarNet has the lowest computational demand among recent SoTA models (using ResNet-based backbone) without a substantial drop in performance, as shown in Figure 12.



**Figure 12: Comparison of SoTA Pillar-based 3d object detectors evaluated on NDS vs. GFLOPs on nuScenes val set. Our Dense Backbone adapted on PillarNet, DensePillarNet, has the lowest computational cost. Also, in comparison with other pillar-based SoTA models, there is not a substantial drop in detection accuracy.**

Figure 13 presents a class-wise performance comparison between our model and the *base* model on the nuScenes dataset. DenseCenterPoint achieves superior performance in most classes, with a ~2% improvement in mAP and a slight increase in NDS over CenterPoint. Although DensePillarNet does not outperform the *base* model, its performance is similar across most classes and there is just ~1.5% drop in NDS. Considering the gains achieved in computational load, the drop in NDS is not substantial.

Method	Car	Truck	Bus	Trail.	CV	Ped.	MC	Bic.	TC	Bar.	mAP	NDS
CenterPoint-Second [7]	<b>83.9</b>	<b>49.5</b>	<b>61.9</b>	34.1	<b>28.3</b>	76.9	44.1	18.0	54.0	59.1	49.4	59.8
DenseCenterPoint	82.1	45.0	50.4	<b>42.9</b>	18.3	<b>77.9</b>	<b>48.9</b>	<b>18.5</b>	<b>64.8</b>	<b>63.6</b>	<b>51.2</b>	<b>60</b>
PillarNet-18 [24]	<b>87.4</b>	<b>56.7</b>	60.9	<b>61.8</b>	<b>30.4</b>	<b>87.2</b>	<b>67.4</b>	<b>40.3</b>	<b>82.1</b>	<b>76.0</b>	<b>65.0</b>	<b>70.8</b>
DensePillarNet	86.9	55.8	<b>65.0</b>	61.0	23.4	85.6	63.6	36.6	80.0	73.7	63.13	69.4

**Figure 13: Evaluation on nuScenes *test* dataset. Additionally, we show the mean Average Precision (mAP) over all classes and the benchmark metric NDS. Abbreviations: Trail.: Trailer, CV: Construction Vehicle, MC: Motorcycle, Ped.: Pedestrian, Bic.: Bicycle, TC: Traffic Cone, Bar.: Barrier.**

Figure 14 presents the results of DensePointPillars on the KITTI *test* set. Our model achieves an improvement of 1-2% in detection accuracy on all classes for 3D as well as BEV tasks compared to the base model. In addition to the promising results, our model requires 33% fewer computations and has four times fewer parameters compared to the base model. Although DensePointPillars has a slightly higher inference time, it still achieves a speed of more than 52 Hz which is more than twice as fast as LiDAR's operating speed of 20 Hz.

Method	mAP↑	Car 3D AP (R40)↑	Ped. 3D AP (R40)↑	Cyc. 3D AP (R40)↑	Time↓ (ms)
		Easy Mod. Hard	Easy Mod. Hard	Easy Mod. Hard	
3D object detection					
PointPillars [14]*	55.44	83.51 73.13 68.02	39.38 32.10 29.54	70.51 54.47 48.34	<b>16</b>
DensePointPillars	<b>57.62</b>	<b>84.60 75.46 68.43</b>	<b>42.76 35.38 32.63</b>	<b>70.95 57.36 50.98</b>	<b>19</b>
BEV object detection					
PointPillars [14]*	63.31	91.25 85.96 80.82	45.74 38.07 35.76	<b>75.76</b> 61.70 54.77	<b>16</b>
DensePointPillars	<b>64.47</b>	<b>92.13 86.31 81.12</b>	<b>47.80 40.31 37.49</b>	75.14 <b>63.27 56.70</b>	<b>19</b>

**Figure 14: Comparison of DensePointPillars vs PointPillars on KITTI *test* set for Car, Pedestrian and Cyclist class. AP is calculated for 40 Recall values at different difficulty levels.**

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 6.

**Table 6: Perception Model optimization - Strength and weaknesses**

	Strengths	Weaknesses
Fairness	The model is plug-and-play and can be adapted on any point-cloud based 3d object detector.	Not evaluated for camera-based 3d object detectors.
Transparency	n/a	n/a
Accountability	n/a	n/a
Privacy	n/a	n/a

## 3.6 Model Uncertainty

### 3.6.1 Introduction

Autonomous driving requires robust and trustworthy systems, especially in safety-critical scenarios involving diverse challenges like adverse weather and sensor corruptions not seen during training. Current methods use multi-camera setups to build 3D occupancy maps, vital for planning and collision avoidance. Ensuring the reliability of these maps under varied conditions is essential. While research improves datasets and architectures, addressing uncertainties from adversarial conditions or distributional shifts has been overlooked, hindering real-world deployment.

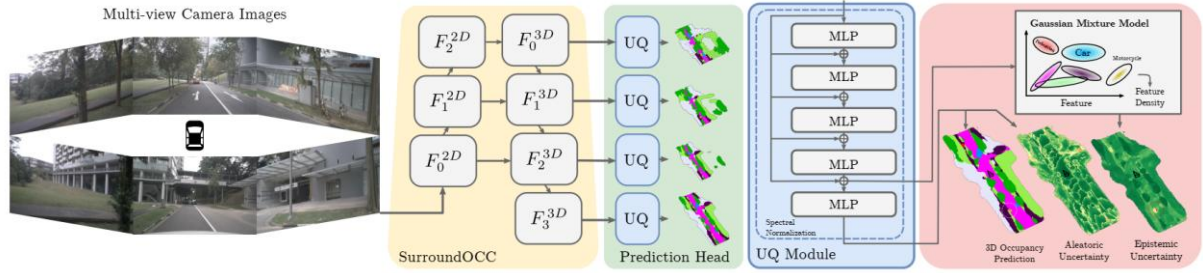
In WP3, IKA successfully developed and adapted an efficient uncertainty estimation method for 3D occupancy prediction, named *OCCUQ* [18]. By integrating a lightweight uncertainty module and training a separate GMM, IKA successfully disentangled aleatoric and epistemic uncertainty. The method was evaluated for Out-of-Distribution (OoD) detection against camera corruptions, including simulated region-specific defects. IKA confirmed findings by inspecting object uncertainty and successfully used epistemic uncertainty for dynamic confidence calibration. Results show *OCCUQ* provides reliable uncertainty measures at scene and region levels, achieving superior performance across downstream tasks like OoD detection and confidence calibration.

### 3.6.2 Methodology

The successfully developed uncertainty quantification (UQ) method for 3D occupancy prediction addresses the critical challenge of uncertainty estimation in autonomous driving perception.

As the base model for 3D occupancy prediction from multi-camera surround view images, the SurroundOCC model was successfully leveraged and adapted. A key achievement was the successful implementation of a novel, lightweight uncertainty module, inspired by Deep Deterministic Uncertainty (DDU). This module was integrated to replace the base model's original prediction head, as shown in Figure 11. The developed module enables efficient and effective uncertainty estimation at the voxel level, providing a crucial capability to successfully disentangle aleatoric and epistemic uncertainties.

The developed system processes input consisting of six multi-view camera images from the vehicle's surround. These images are passed through a feature backbone to generate a dense feature volume. The custom uncertainty module then processes this volume. Beyond predicting voxel occupancy and semantic class, this module successfully integrates a Gaussian Mixture Model specifically designed to predict the model's epistemic uncertainty. The final output of the developed method provides a comprehensive set of predictions: occupancy and semantic class for each voxel, alongside the corresponding estimated aleatoric and epistemic uncertainties. This successfully implemented methodology provides the necessary uncertainty information to enhance the trustworthiness and interpretability of the perception system.



**Figure 11: From multi-view camera images our method provides 3D occupancy predictions with reliable epistemic at voxel level. We build on top of the SurroundOCC model and introduce an Uncertainty Quantification (UQ) module.**

### 3.6.3 Results

The effectiveness of the developed uncertainty quantification approach in capturing epistemic uncertainty was evaluated using the nuScenes and MultiCorrupt datasets. An Out-of-Distribution (OoD) detection framework was established at both scene and region levels. In this evaluation context, the developed approach consistently outperformed baseline methods such as Monte Carlo Dropout (MCD) and Deep Ensembles (DE) on standard metrics like AUROC and FPR95.

Method	nuScenes		MultiCorrupt		Scene Corruptions		Region Corruptions		Params↓	Time(s)↓	Memory↓
	IoU↑	mIoU↑	IoU↑	mIoU↑	mAUROC↑	mFPR95↓	mAUROC↑	mFPR95↓			
MCD <sub>n=5</sub>	0.331	0.209	0.242	0.137	63.03	87.44	51.21	94.45	<b>180.07M</b>	2.11	<b>7.1GB</b>
DE <sub>n=3</sub>	<u>0.338</u>	<u>0.218</u>	<u>0.246</u>	<u>0.145</u>	68.46	76.64	54.26	94.16	540.22M	1.26	21.3GB
DE <sub>n=5</sub>	<b>0.341</b>	<b>0.221</b>	<b>0.247</b>	<b>0.148</b>	69.58	74.89	55.26	94.23	900.37M	2.11	35.5GB
Entropy	0.328	0.208	0.242	0.141	54.12	95.06	50.01	94.32	<b>180.51M</b>	<b>0.42</b>	<b>7.1GB</b>
Max. Softmax	0.328	0.208	0.242	0.141	55.42	87.71	50.22	95.65	<b>180.51M</b>	<b>0.42</b>	<b>7.1GB</b>
Ours	0.329	0.210	0.242	0.141	<b>80.17</b>	<b>56.39</b>	<b>57.81</b>	<b>93.60</b>	<u>180.51M</u>	<u>0.45</u>	<u>7.1GB</u>

**Figure 12: Quantitative evaluation of our proposed method against the established baselines. We report accuracy metrics (IoU and mIoU), epistemic uncertainty quality proxies (mAUROC and mFPR95) and model complexity indicators. mAUROC and mFPR95 are computed across all image corruptions and all severity levels.**

Furthermore, the epistemic uncertainty estimates, indicated by the log density, demonstrated effective sensitivity to different types and severities of corruption. As corruption severity increased, the feature density showed improved separation between in-distribution and out-of-distribution data, successfully quantifying epistemic uncertainty by assigning higher uncertainty values to unseen data. Overall, the results indicate that the efficient uncertainty estimation technique provides reliable uncertainty measures at both scene and region levels and achieves superior performance across related downstream tasks, including OoD detection and confidence calibration.

In conclusion, the OCCUQ successfully provides reliable epistemic uncertainty estimates for 3D occupancy prediction, outperforming baselines and demonstrating effectiveness across various corruption scenarios and downstream tasks.

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 7.

**Table 7: Model uncertainty - Strength and weaknesses**

	<b>Strengths</b>	<b>Weaknesses</b>
<b>Fairness</b>	Does consider class imbalances or rare objects / rare object variances by marking it with higher uncertainty.	Does not explicitly model ethic variances, but rather does it implicitly by the give training data.
<b>Transparency</b>	Gives fine grained model uncertainty, which also fosters transparency.	Does not give explicit explanation for detection and uncertainty.
<b>Accountability</b>	N/A	N/A
<b>Privacy</b>	N/A	N/A

## 3.7 LiDAR-Radar-Camera Fusion

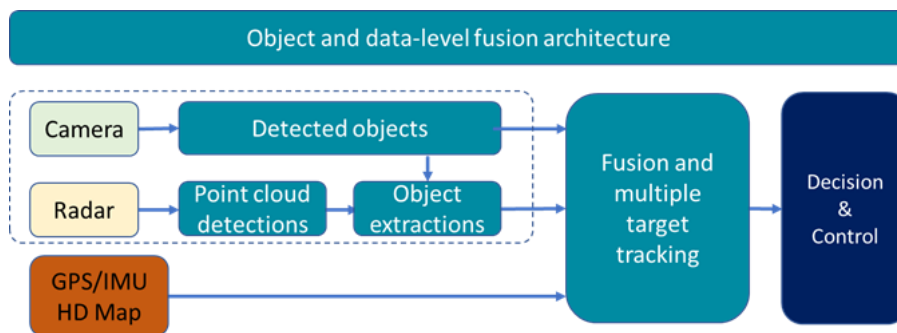
### 3.7.1 Introduction

Autonomous vehicles rely heavily on fault tolerance and detection mechanisms to ensure safe and reliable operations. Cameras are more efficient in determining the features of an object and are hence employed for functionalities such as understanding traffic signs, detecting and classifying objects. Radar sensors can capture well the motion characteristics of an object with high resolution, whereas LiDAR has wide coverage for detection along with superior ranging performance. Cameras are vulnerable to varying illumination conditions, and LiDAR is erroneous under extreme weather conditions, whereas radar is more robust under adverse weather, environmental, and illumination conditions. The advantages of each type of sensor are utilized by using the concept of sensor fusion. Sensor fusion collectively processes inputs from various sensors and derives an interpretation of the environment surrounding the vehicle with a higher level of certainty.

### 3.7.2 Methodology

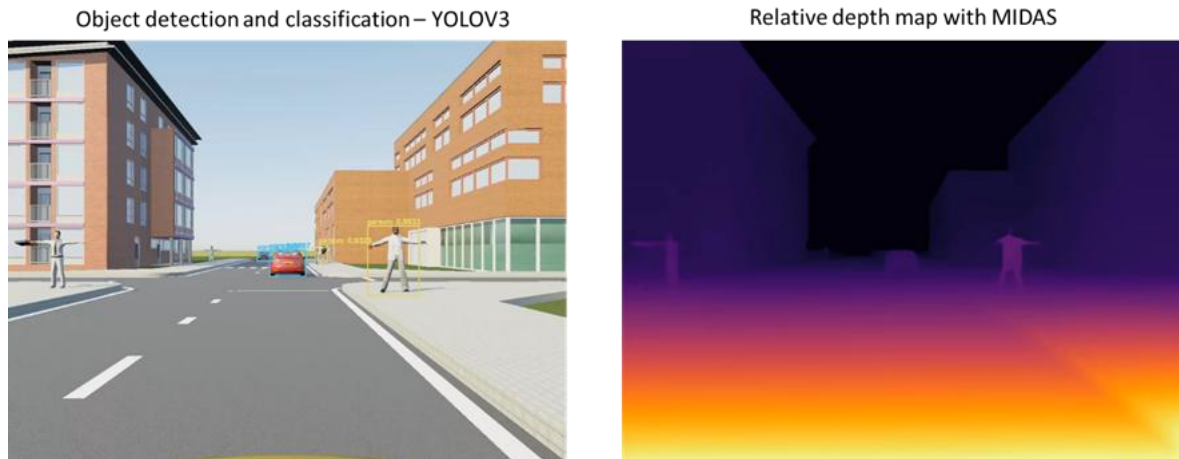
Building upon the work presented in D3.2, SIE-BE dedicated the remaining reporting period on enhancing the quality of its pseudo-label generation algorithm. This was achieved by adopting a more advanced fusion strategy and integrating additional pretrained 3D LiDAR detectors compared to previous efforts in D3.2. The diverse 3D detection proposals from both LiDAR and Radar detectors are subsequently combined using a Kernel Density Estimation function, resulting in more precise pseudo-labels. More specifically, the method fused the centroids ( $center_x$ ,  $center_y$ ,  $center_z$ ), the orientation, and the confidence scores of the object lists for each dimension separately, leading to more accurate localization of the objects using the available sensor modalities.

The sensor fusion methodology developed by SIE-NL has been tested using the generated synthetic data, where the architecture of the sensor fusion system is illustrated in Figure 13.



**Figure 13: Camera and data fusion to reduce conflicting perception**

The camera sensing and perception stack usually has two main components – object detection and classification using e.g. YOLOV3 as well as a depth estimation component like MIDAS, see Figure 14.

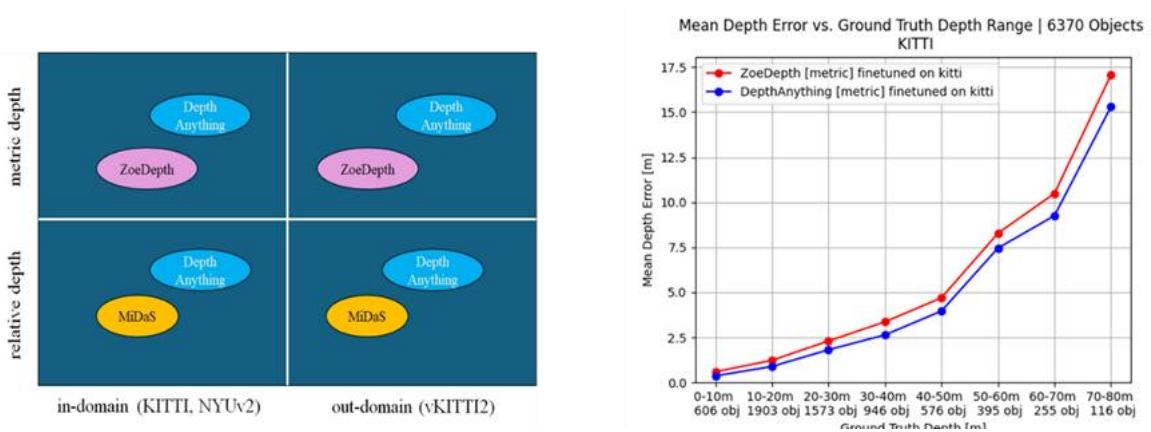


**Figure 14: Camera sensing and perception stack with two open-source components**

SIE-NL investigated and compared different open-source depth estimation algorithms such as: MIDAS, ZoeDepth and DepthAnything. Some of these algorithms are only for relative depth estimation other can be used also for metric depth estimation after fine-tuning the network.

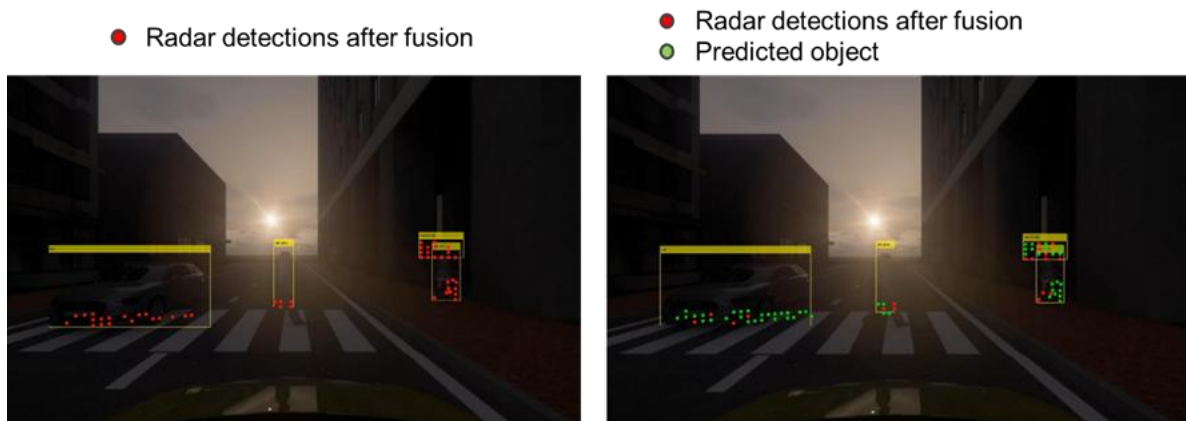
### 3.7.3 Results

SIE-NL investigations show that open-source monocular depth estimation algorithms performance is somewhere around 10% absolute relative error – see Figure 15. This error level might be too high for automotive applications, therefore in terms of the depth estimation, we relied on the depth detected using the radar sensor.



**Figure 15: Depth estimation algorithms and their performance**

In addition to the sensor fusion performed between the camera and radar sensor, we also added a simple prediction, which improves the robustness of detections, as shown in Figure 16. The simple prediction mentioned earlier is one time-step (0.05 seconds) ahead prediction in time for the target, considering a simple kinematic model. This improves the robustness of detections, since target detection can be missing at certain time moments and present at the next time moment.



**Figure 16: Camera and radar sensor fusion (left) enhanced with prediction (right)**

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 8.

**Table 8: LiDAR-RADAR-Camera Fusion - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	Dual-channel architecture, easy fault detection, safe approach	Higher complexity and difficulty in the implementation
<b>Transparency</b>	Object level sensor fusion, transparent implementation for the user	Camera-based object detection is based on deep neural network.
<b>Accountability</b>	Dual-channel architecture with divers sensing, fault-detection and fault-isolation	Common-cause fault could happen, but probability is very low and is considered during design
<b>Privacy</b>	No privacy related issues, since synthetic data is used.	None

## 3.8 LiDAR-Camera Fusion

This development by CAF has not generated more results than those reported in D3.2. The efforts from this task have been reallocated in the development of a new Radar-Camera fusion and the development of a deliverable and reusable version of all algorithms (this one included).

## 3.9 Explainable Object Level Fusion

### 3.9.1 Introduction

Autonomous mobility relies on accurate perception for safe navigation, but individual sensor modalities often fall short. Sensor fusion addresses this by integrating data from multiple sensors, reducing limitations, and improving accuracy.

A robust Multimodal and Multi-Object Tracking (MOT) [19] method developed by IDIADA is applied on top of unimodal detections generated with AI models to increase perceptual robustness and manage conflicting perceptions. AI-based model outcomes are difficult to interpret and explain, while the MOT by design allows to make the data fusion more explainable and traceable. Explainability is achieved through visualization of detections, allowing users to identify which sensor is responsible for each detection. Moreover, the system collects data fusion statistics to capture and analyse potential perception conflicts.

### 3.9.2 Methodology

Improving the transparency and trustworthiness of multi-sensor fusion requires tools to detect, quantify, report, and visualize conflicts in perception. Additionally, it is essential to assess the contribution of each sensor modality to the final fused result. The methodology presented here is implemented within IDIADA's Kalman Filter-based MOT framework, which supports fusion of asynchronous and heterogeneous inputs from cameras, radars, and LiDARs.

Perception conflicts are detected by analysing the innovation vector (or residual) produced by each sensor. For a given sensor  $i$  and time  $k$ , the innovation is defined as the difference between the actual measurement and the predicted observations  $r_k^i = z_k^i - H^i \bar{x}_k$ , where  $z_k^i$  is the measurement,  $\bar{x}_k$  is the predicted state, and  $H^i$  is the observation matrix. A large innovation norm indicates a significant mismatch between prediction and observation, suggesting a potential conflict.

Visualization plays a key role in making conflict analysis interpretable. Our approach includes graphical overlays of individual sensor detections and the final fused track, providing insight into the alignment or discrepancy between modalities. We represent innovation vectors graphically to show the magnitude and direction of disagreement between predicted and observed measurements. Covariance ellipses are used to illustrate the uncertainty around each object's position, offering further insight into the system's confidence in the fusion.

To quantify these conflicts, we define a conflict score per tracked object at each time step  $k$  from  $N$  contributing sensors as:  $C_k = \sum_i^N (r_k^i{}^T S_k^{i-1} r_k^i)$ , where  $S_k^i$  is the innovation covariance matrix. By tracking the frequency of high conflict scores (e.g., the percentage of frames where a threshold is exceeded), the system can assess the temporal consistency of perception conflicts.

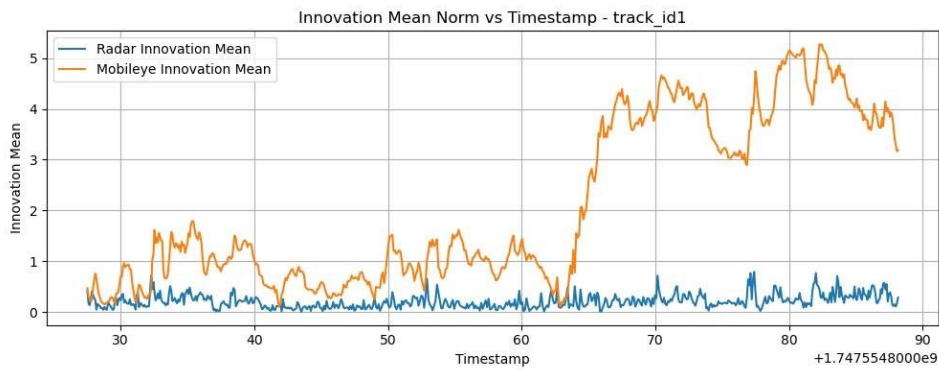
Understanding the individual contribution of each sensor modality to the fused track is critical for explaining and validating the fusion process. This is achieved by analysing the Kalman gain matrix  $K$

associated with each sensor update, which determines how strongly a given measurement influences the state estimate. Specifically, the state correction term, expressed as  $\Delta \bar{x}_k^i = K_k^i r_k^i$  quantifies the adjustment made to the predicted state  $\bar{x}_k$  based on the innovation  $r_k^i$  from sensor  $i$ . By examining the magnitude  $\Delta \bar{x}_k^i$ , we can assess the degree to which each sensor contributes to the fused result at each time step.

### 3.9.3 Results

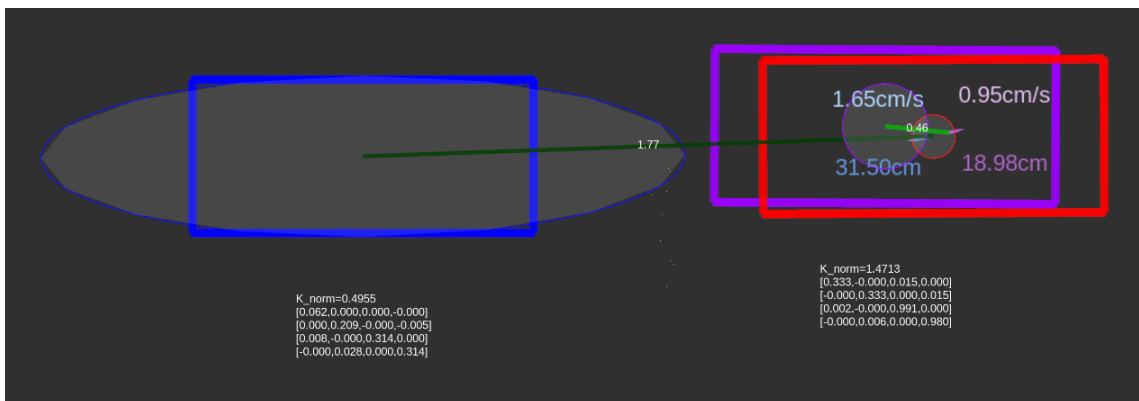
To evaluate perception conflicts and sensor contributions within IDIADA’s Kalman Filter-based MOT framework, we analysed innovation mean, conflict score, and sensor influence across different driving scenarios.

Figure 17 illustrates the mean innovation norm over time for two sensors: radar and Mobileye (camera). The Mobileye sensor shows a marked increase in innovation norm at certain time intervals, indicating a discrepancy between its measurements and the predicted state. This suggests a potential perception conflict, as per the methodology where larger innovations highlight mismatches between prediction and observation.



**Figure 17: Innovation mean over time for radar and Mobileye per tracked object.**

Figure 18 illustrates a perception conflict where the Mobileye detection (blue) diverges from both the radar detection (purple) and the fused track (red). Covariance ellipses show detection uncertainty, and a green line—brightened by Kalman gain norm—represents the Mahalanobis distance between detections. Additionally, each sensor’s Kalman gain matrix is visualized below its detection. The state correction vectors (position and velocity)—color-coded by sensor—and their magnitudes are shown to highlight the effect of each measurement on the final estimate.



**Figure 18: Mobileye-track conflict detection.**

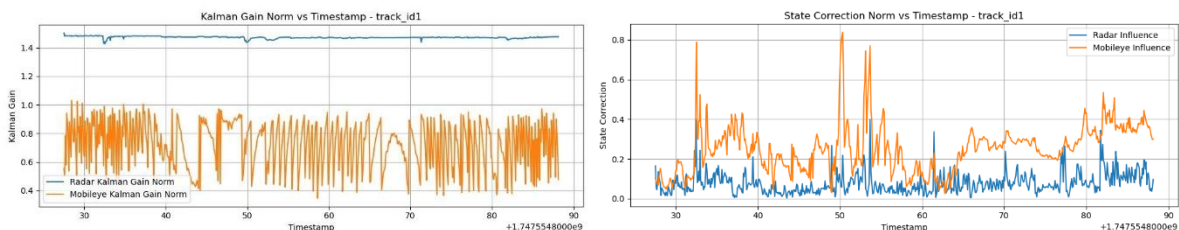
To assess how often perception conflicts occur over time, Table 9 reports the percentage of frames in which the conflict score exceeds various thresholds, across different driving scenarios (e.g., followed car, cut-out, and cut-in).

**Table 9: Percentage of frames where conflict score (C\_k) exceeds thresholds by scenario type.**

Scenario Type	C_k >1	C_k >2	C_k >3	C_k >4
Followed car	39.27%	26.50%	8.04%	0%
Cut-out car	30.56%	26.56%	25.22%	23.89%
Cut-in car	3.04%	0%	0%	0%

These results show how scenario dynamics affect conflict frequency. In the Followed Car scenario, moderate conflicts arise from relative velocity changes that the constant velocity Kalman model struggles to capture. The Cut-out scenario shows high conflict rates, especially at long ranges (>80 m), where Mobileye’s uncertainties are larger. In contrast, the Cut-in scenario shows minimal conflicts, consistent with short distances (<35 m) where longitudinal estimates are more accurate.

To illustrate each sensor’s influence on the fused state, Figure 19 presents the evolution of the Kalman Gain Norm (left) and State Correction Norm (right) over time for a representative track. The Kalman Gain Norm reflects the trust assigned to each sensor: Radar maintains a consistently high gain (~1.45), indicating stable reliability, while Mobileye shows greater variability, reflecting fluctuations in measurement quality. The State Correction Norm quantifies how much each sensor adjusts the predicted state—Radar’s small, consistent corrections suggest good alignment with predictions, whereas Mobileye’s larger corrections indicate more frequent deviations.



**Figure 19: Kalman gain and State correction per sensor over time for a specific track.**

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 10.

**Table 10: Explainable object level fusion - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	Sensor contributions can be quantified, promoting equitable decision logic. Reduces overreliance on a single source.	Fusion outcomes may still be biased if sensor coverage is imbalanced.
<b>Transparency</b>	Visualization of sensor detections and their influence on tracked objects (e.g., Kalman gain, correction vectors, ellipses) enhances interpretability.	Requires expert knowledge to fully understand statistical conflict indicators.

<b>Accountability</b>	Conflict logging and per-sensor influence tracking enable traceability per frame.	Real-time auditability may be limited under high-load conditions.
<b>Privacy</b>	No personal data is stored; the system operates with processed sensor data (position, velocity).	N/A

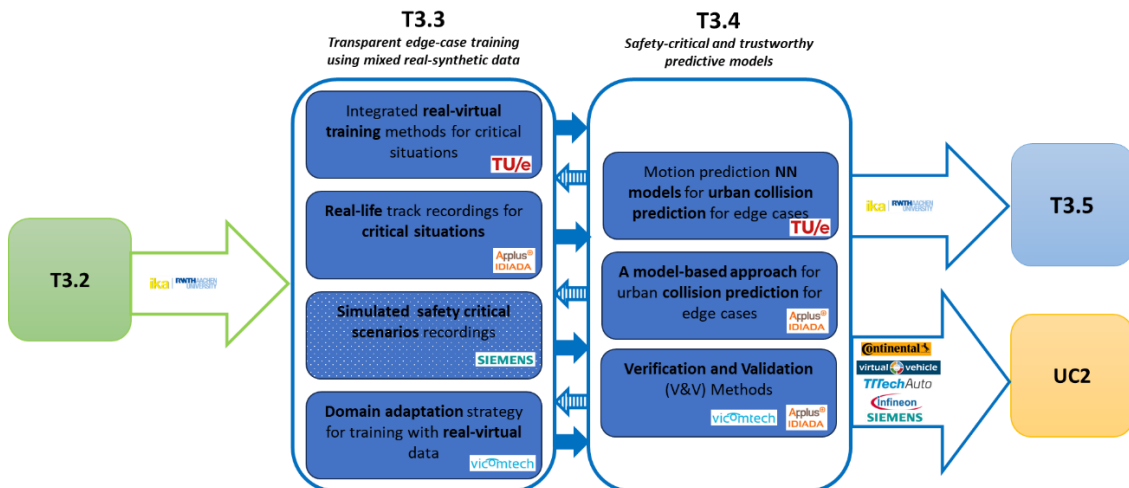
### 3.10 Local Dynamic Maps

This development by VIC has not generated more results than those reported in D3.2. The efforts from this task have been reallocated in the development of a scene understanding algorithm for U.C 2.1, presented in Section 4.4.

## 4 TRANSPARENT EDGE-CASE TRAINING USING MIXED REAL-SYNTHETIC DATA AND SAFETY-CRITICAL AND TRUSTWORTHY PREDICTIVE MODELS

### 4.1 Introduction

T3.3 Transparent edge-case training using mixed real-synthetic data and T3.4 Safety-critical and trustworthy predictive models are heavily intertwined, as also shown in Section 1.3. Therefore, in this chapter the work on the development of AI algorithms that are related to those topics is combined. Figure 20 provides the approach and dependencies between the different modules that have been developed so far. Specifically, for use case 2.2 (further noted as UC2.2), work was allocated to develop dedicated algorithms to demonstrate UC2.2 at a later stage in the project. The work done by the partners involved in that use case is therefore explicitly mentioned in this chapter.



**Figure 20: Approach and dependencies between T3.3, T3.4, and specifically UC2.2**

The objectives of the algorithms developed in this chapter are to

1. improve data efficiency and robustness of the predictive models using mixed real-synthetic data to train deep neural networks for edge cases
2. increase the explainability and trustworthiness of the predictive models using hybrid AI techniques that provide the closest physical explanation to the prediction of a deep data-driven approach

## 4.2 Collision risk prediction

### 4.2.1 Introduction

Road accidents are still a major concern of both the automotive industry and society, as 1.35 million people worldwide lose their lives in traffic every year [20]. Collision Risk Prediction (CRP) is essential in preventing accidents: [21] shows that a 1.5-second warning can prevent 90% of crashes for human drivers, with even greater potential for autonomous systems. For autonomous vehicles, this probability could even be higher due to faster reaction times. The task of detecting a future collision before it happens is intrinsically challenging, mainly due to the huge variety of accident scenarios, and the difficulty of creating a well-annotated big dataset covering all those scenarios. Dashcam-based CRP is a method for assessing the risk of a future collision using real-time dashcam footage.

This task contributes:

- A literature review of CRP models (including our own initial RiskNet (see D3.2 [1])) and datasets.
- Reproduction and improvement of the Dynamic Spatial-Temporal Attention (DSTA) model [22].
- Integration of optical flow into DSTA to enhance predictive performance.

### 4.2.2 Methodology

#### Model Evaluation

A broad range of CRP models was evaluated. Early models like **DSA** [23] used spatial attention in RNNs. More advanced models such as **UString** [24] and **DRIVE** [25] introduced graph neural networks and reinforcement learning, respectively. **RiskNet** [26] trained exclusively on synthetic videos (PreScan simulator) using optical flow and safe/unsafe labels. However, **RiskNet was unsuitable for real-world datasets** because it:

- relied on **non-standard per-frame labels** (safe/unsafe), absent from common CRP datasets.
- performed poorly when transferring from synthetic to real-world data.
- RiskNet's reliance on its own datasets (PreScan, YDID) further limited its applicability to standard benchmarks.

In contrast, **DSTA** [22] used temporal and spatial attention with GRUs, achieving state-of-the-art results on real-world datasets (CCD, DAD). Its architecture was relatively simple, reproducible, and supported with a well-maintained GitHub repository.

#### Dataset Evaluation

Popular datasets evaluated include:

- **CCD** [24]: large (4500 videos), high annotation quality, but performance on it is saturated (most models achieve mAP 99–99.9%).

- **DAD** [27]: smaller (1750 videos), more challenging, with lower reported mAPs (~50–77%).
- **DADA-2000** [28], **A3D** [29], **DoTA** [30], and **Nexar** [31] add variety but vary in annotation detail and availability.

**Model Selection:** Based on the literature review, DSTA was chosen for:

- demonstrated strong performance on CCD/DAD.
- standard architecture (CNN + GRU + attention).
- availability of public code and reproducibility.
- compatibility with real-world datasets.

Finally, recent models like THAT-net [32] and RiskNet had shown that **integrating optical flow** improves performance by capturing motion cues critical for CRP. Thus, an additional goal was to extend DSTA with optical flow.

## Methodology

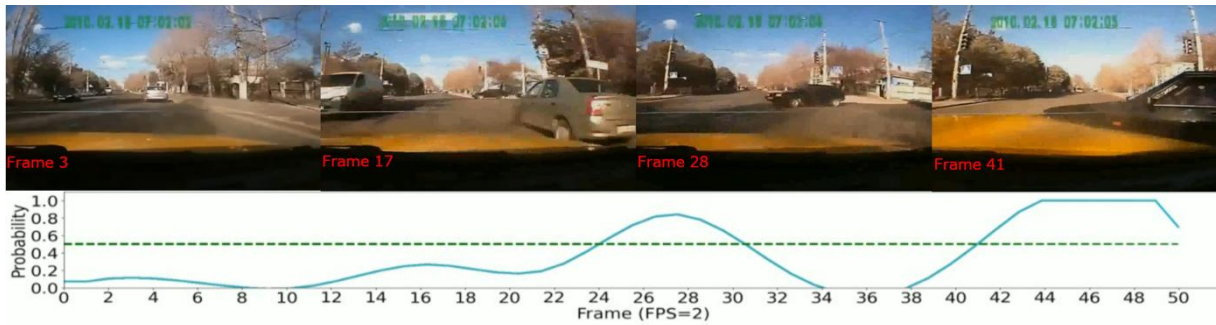
Based on the literature research, earlier experience with RiskNet (and its limitations) and optical flow, we continued our research by:

1. **Reproducing original work:** Updated DSTA’s code for compatibility with modern libraries. Replaced the original KITTI-trained object detector with a COCO-trained Cascade R-CNN, requiring label remapping.
2. **Retraining and Testing:** Trained DSTA on both **CCD** and **DAD** datasets. The original pre-trained weights were found unreliable on CCD; training from scratch reproduced the expected performance.
3. **Improvement (Optical Flow):** Extended DSTA by adding optical flow as input, extracted with **RAFT** [33]. Two integration strategies were tested:
  1. **Zero-Padding:** padding missing detection slots.
  2. **ROI Duplication:** duplicating existing object features to preserve model structure.

### 4.2.3 Results

On CCD, our trained DSTA reproduced the original paper’s results (AP 97.58%, mTTA 4.95s). Figure 21 shows a qualitative output of DSTA on a CCD video.

The GitHub pre-trained weights performed poorly (AP 68.18%), validating the need for retraining.



**Figure 21: DSTA on CCD dataset, showing several individual frames (top) and the output of the model (probability of a collision) (bottom)**

On DAD, adding optical flow with ROI duplication substantially improved both accuracy (AP 80.34%) and earliness (mTTA 2.53s)—outperforming both the original DSTA and our RGB-only version. Zero-padding improved AP substantially but sacrificed mTTA (see Table 11).

**Table 11: Performance comparison of improved DSTA-Flow against original DSTA**

Experiment	AP [-]	mTTA [s]
<i>Original DSTA</i>	72.34	1.5
<i>Our trained DSTA</i>	70.08	2.33
<i>DSTA-Flow (Zero Padding)</i>	74.85	1.66
<i>DSTA-Flow (ROI duplication)</i>	<b>80.34</b>	<b>2.53</b>

While RiskNet inspired the use of optical flow, its reliance on synthetic data and incompatible labelling made it unsuitable for real-world CRP tasks. DSTA provided a robust and practical foundation. By extending DSTA with optical flow, we leveraged motion-based features to achieve state-of-the-art performance improvements—demonstrating the value of combining spatial, temporal, and motion cues in CRP.

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 12.

**Table 12: Collision Risk Prediction - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	Can be trained on all sorts of classes and is not dependent on classical object tracking models (such as Kalman Filters), providing usefulness to a wider range of detected objects	Applicability depends on training data. Edge cases (mainly difficult scenarios) are still unclear.
<b>Transparency</b>	Can indicate the probability of a collision well ahead in time	Does not give explicit explanation for detection and uncertainty. Neither does the probability always ensure a correct collision prediction.
<b>Accountability</b>	N/A	N/A
<b>Privacy</b>	By including optical flow, we decrease the reliance on privacy sensitive data and can rely more on motion data instead. Thereby improving privacy.	N/A

## 4.3 Safety-critical and trustworthy prediction models

### 4.3.1 Introduction

The latest AI development has proved to be a powerful tool to analyse and assess risks while driving. In this context, the availability of camera recordings of potentially dangerous scenarios allows us to pre-emptively prevent collisions. However, data-based approaches lack soundness and explainability, thus there's resistance of both institutions and public to introduce them in critical modules of a vehicle. To improve confidence on the algorithms, the state of the art is developing towards explainable AI, which aims to derive the reasoning behind a neural network output. Some examples can be found in Saliency maps [34], which highlight what regions of an input image are given more attention by the network, and the SHAP algorithm [35], used to evaluate what input variations lead to changes in the output.

### 4.3.2 Methodology

A novel AEB trigger prediction network has been developed in collaboration with TU/e. Using a transformer architecture to predict a binary output from a sequence of frames, the network is trained to decide whether an AEB should trigger considering only a camera feed as the input of the system. A state-of-the-art Time-To-Collision (TTC) dataset [36] has been reannotated with Boolean labels, being 0 in normal driving conditions and 1 in the case where an industry standard AEB, presented in [37], would be triggered. We used a driver-warning criteria, where whenever the TTC is smaller than 3 seconds a warning sign is triggered.

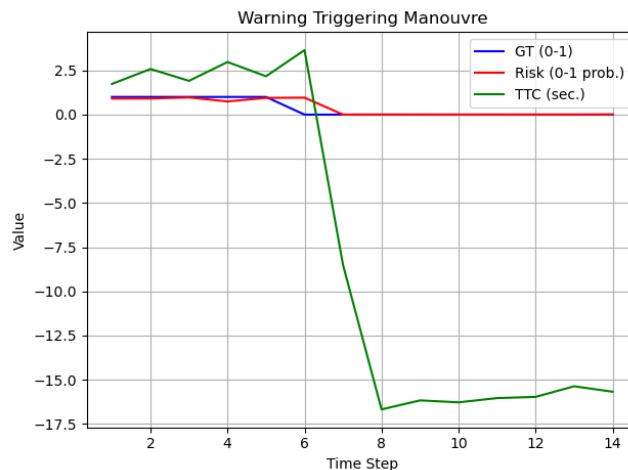
### 4.3.3 Results

An initial overview of the algorithm performance using a validation set of 33360 samples shows how, even though the overall accuracy is high, the system presents a low precision rate as presented in Table 13. Such behaviour highlighted a major flaw in the dataset, since it's mostly conformed of data in regular driving conditions, leading to a poor representation of potentially dangerous scenarios. When evaluating the same metrics over a randomly sampled balanced validation subset, we can observe an improvement over the true positive rate but a richer validation testbench is required to extract solid conclusions.

**Table 13: Algorithm Performance Metrics**

	Accuracy	Precision	Recall	F1 Score
Full validation set	0.96	0.55	0.90	0.68
Balanced validation set	0.93	0.95	0.90	0.93

In Figure 22, it can be seen how the algorithm behaves in the face of a dangerous scenario, where the ego and target vehicles become close enough to send a warning signal to the driver. Y



**Figure 22: Algorithm evaluated in a positive sample of the dataset**

In conclusion, this initial approach to generating an AI-based AEB is promising; however, the nature of the data used to develop the algorithm is limited to extract solid conclusions regarding its scalability in real scenarios. Further research needs to be conducted in terms of data acquisition and benchmarking with traditional implementations to evaluate whether an AI algorithm can outperform an AEB in complex situations, where the context of the scene could give additional information to the prediction network, not available to traditional AEB algorithms.

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 14.

**Table 14: Collision Risk Prediction - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	Only requires a camera feed, which is widely available in modern vehicles	Does not consider vehicle and environment information. The dataset used is not well balanced, which may cause misleading results
<b>Transparency</b>	Input-output relation is simple and understandable by the user	No additional information on how the network processes the scene
<b>Accountability</b>	The algorithm does not take control over the system	Warning signals might be overlooked by the driver
<b>Privacy</b>	Subject to training data	Subject to training data

## 4.4 Context understanding through VLM

### 4.4.1 Introduction

Vision Language Models (VLMs) are advanced neural network algorithms that integrate text and image processing, enabling tasks such as image captioning, visual question answering, and image classification. Trained on massive amount of data, VLMs possess a deep understanding of both visual content and human language, allowing them to generalize well and perform zero-shot learning. Their ability to semantically interpret images makes them suitable for complex tasks like scene

understanding in autonomous driving. This project aims to explore VLMs within the Connected, Cooperative, and Automated Mobility (CCAM) context to develop AI-powered components that analyse multi-sensor data and generate semantic descriptions of driving environments, highlighting safety-critical situations for downstream decision-making processes such as trajectory planning and V2X communication.

#### 4.4.2 Methodology


To evaluate how effectively VLMs can be applied in traffic scenarios, we use GPT-4o, a state-of-the-art VLM, to analyse multiple driving environments. The model receives multiple images as input, specifically from the vehicle’s front camera and a bird’s-eye view (BEV) perspective, though additional sensor data could be integrated if needed. To standardize the output, GPT-4o is instructed to generate a structured JSON file containing predefined fields. These fields capture critical scene elements such as the presence and location of pedestrians and bicycles (either on the road or sidewalk), the status of traffic lights (semaphores), and a risk assessment including a description of the potential hazard. This structured output can then be directly used to trigger relevant icons, alerts, or alarms in our visual interface, enhancing situational awareness and explicability in real time.

### Traffic Scene Analysis Gallery

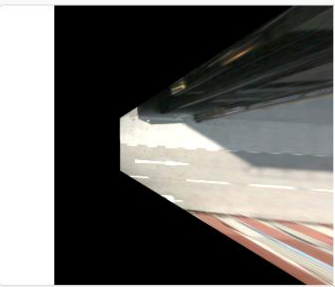
< Previous 3 of 8 Next > Return to Analysis

**Scene Images**


Front View




BEV View




**Scene Elements**




Person in Road




Pedestrian on Sidewalk



Bicycle in Road



Bicycle on Sidewalk



Semaphores: green

**⚠ Risk Detected**

Person near truck in road, potential collision risk and traffic obstruction.

**Thought Process**

**Analysis of the Scene:**

- Identifying Elements:**
  - There is a large truck parked on the left side of the road.
  - A person is standing near the truck, possibly unloading or loading items.
  - Multiple vehicles are on the road, moving in both directions.
  - There are traffic cones or barriers on the right side of the road.
  - A traffic light is visible in the distance.
- Relationships and Positions:**
  - The person near the truck is on the edge of the road, potentially in a hazardous position.
  - The truck is occupying part of the road, which may affect traffic flow.
  - Vehicles are moving past the truck,

**Figure 23: Visualization of the context understanding implementing VLMs. The images are the input to the model, with instructions to identify potential risks. As response, the model must alert if there are any person in the road, sidewalk, any bicycles on the road or sidewalk, and the state of the semaphores if any. Also, the thought process is shown on the side.**

### 4.4.3 Results

We tested our tool and found that VLMs like GPT-4o can effectively understand and interpret various types of traffic data. Thanks to their strong visual reasoning and built-in traffic knowledge, these models can detect elements such as pedestrians and traffic lights and even identify potentially dangerous traffic situations. Remarkably, this capability works out-of-the-box without any additional training, while also allowing the model to classify and recognize a wide range of scenarios by leveraging its own internal knowledge to interpret complex scenes. On the other hand, we observed that for GPT-4o to accurately interpret an image, key elements like pedestrians need to be clearly visible and not too distant from the camera. Moreover, we can display in the tool the response of GPT-4o for the user to know how the model is interpreting the image, as represented in Figure 23.

	Strengths	Weaknesses
<b>Fairness</b>	N/A	For this research, GPT-4o model has been used. There is some undisclosed data regarding this model that do not allow for knowing the data it was trained with and its representativeness.
<b>Transparency</b>	With the thought process, more descriptions of the scene and details of how the model gets to a conclusion is provided. This can help people know what could have been the logic of the model.	N/A
<b>Accountability</b>	N/A	For this research, GPT-4o model has been used. There is some undisclosed data regarding this model that do not allow for knowing the data source.
<b>Privacy</b>	N/A	Privacy issues are not considered in this methodology; an anonymizer should be applied before making the prediction with the image.

## 4.5 Domain adaptation strategy for training with real and synthetic data

### 4.5.1 Introduction

Machine learning systems encounter limitations stemming from enough and relevant training data availability. This challenge is particularly pronounced in edge cases. Synthetic data can help alleviate these data needs if the domain gap between real and synthetic data is correctly handled or mitigated. In domains such as autonomous driving, synthetic data generated through simulation has become a practical solution to augment scarce or hard-to-capture real-world scenarios. However, despite its advantages in scalability and control, a key limitation remains: the domain gap—the distributional differences between synthetic and real data—which can significantly hinder model generalization. Bridging this gap is crucial to ensure that models trained or validated with synthetic data perform reliably in real-world conditions.

### 4.5.2 Methodology

VIC explored the best strategy to train a Deep Neural Network (DNN) using annotated synthetic data (source domain) and some unannotated real data (target domain) in a classification task.

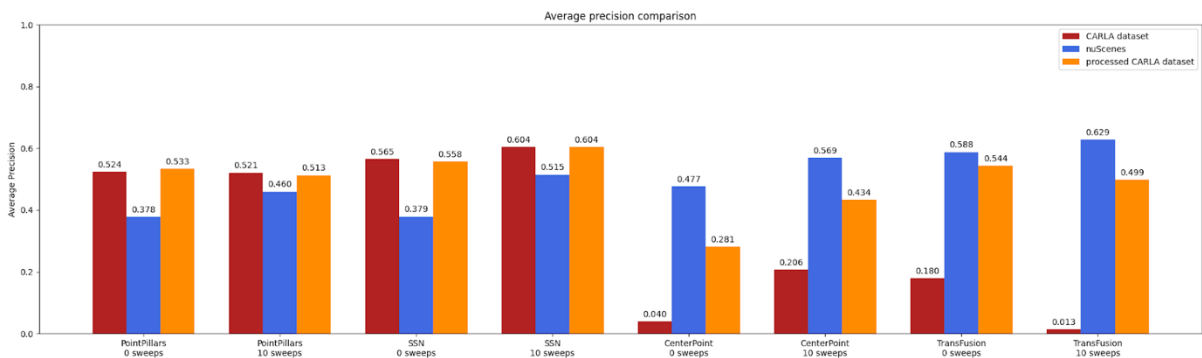
In D3.2, a study about domain gap in images from adverse weather datasets (SHIFT [38] for synthetic data and BBDK100 [39] for real data) was presented. VIC has continued working in domain gap with point cloud datasets.

In the case of point clouds, our work focuses on 3D object detection and the impact of the domain gap when using synthetic data generated with CARLA. The domain gap refers to the distributional differences between synthetic and real-world point cloud data, which can significantly affect the performance of models trained or evaluated across domains. In this study, we explore how state-of-the-art object detection models, trained with real data from nuScenes, behave when tested on synthetic and processed synthetic point clouds, without using any real annotations for training.

The methodology involves evaluating model performance on three types of data: real point clouds, unprocessed synthetic point clouds, and synthetic point clouds modified to better resemble real-world data [40]. The processing step addresses key discrepancies observed in synthetic point clouds, such as unrealistic intensity values, overly regular scan patterns, and the lack of sensor noise. This approach allows us to systematically analyse how domain differences influence detection accuracy and to assess the effectiveness of simple processing techniques in mitigating the domain gap.

### 4.5.3 Results

Figure 24 shows the average precision of four 3D object detection models evaluated on real, synthetic, and processed synthetic point clouds. The processed version includes noise injection, random point removal, and normalization of intensity values to better mimic real sensor data. While PointPillars and SSN perform consistently across all domains, CenterPoint and TransFusion show a significant drop on raw synthetic data. After processing, their performance improves notably—especially with TransFusion—indicating a clear reduction in the domain gap.



**Figure 24: Average precision comparison of four 3D object detection models using real (nuScenes), synthetic, and processed synthetic point clouds (using CARLA), with and without point cloud accumulation. The results highlight the domain gap and the effectiveness of the proposed synthetic data processing.**

These results confirm the existence of a domain gap between real and synthetic point clouds, with its impact varying across model architectures. Simpler models appear more robust, while advanced architectures benefit more from domain adaptation. The proposed processing method narrows the accuracy gap, particularly when using accumulated point clouds, making synthetic data more reliable

for evaluating 3D object detection systems. Future work should explore its effectiveness in training pipelines and investigate additional strategies to further reduce the domain gap.

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 15.

**Table 15: Domain adaptation strategy for training with real and synthetic data - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	With synthetic data, it is possible to add more diversity and more representative data than with real data. This is the first step to build more fair models.	N/A
<b>Transparency</b>	N/A	N/A
<b>Accountability</b>	When producing synthetic data, the source is clear and controllable. This is not always the case with real data, with which sometimes the source is unknown or non-traceable.	N/A
<b>Privacy</b>	Using synthetic data help protect person’s privacy. This methodology aims for reducing the gap between real and synthetic data in a way that the model’s training and performance is not affected by this change of domain.	N/A

## 4.6 AI algorithm for application in UC2.2

The primary objective of UC2.2 is to enhance the robo-taxi's ability to navigate safely through urban traffic by predicting the intended motion of the traffic participants. This goal is achieved through the development of a comprehensive prediction module, which goes beyond a standalone prediction model. The module will incorporate components (see Figure 25) for (1) creating a representative urban scenario dataset, (2) processing raw sensor measurements to detect not only the traffic participants but additional scenario information, (3) generating motion predictions for detected traffic participants, and (4) implementing fault-tolerant decision-making. During the first cycle, the focus was on independent development and optimisation of components ensuring their performance within each specific scope. In the second cycle, we continue to improve individual components, while simultaneously initiating the integration of the results achieved by collaborating partners. Therefore, the following sections will provide a detailed overview of improvements made within individual components and report on the progress toward their integration.

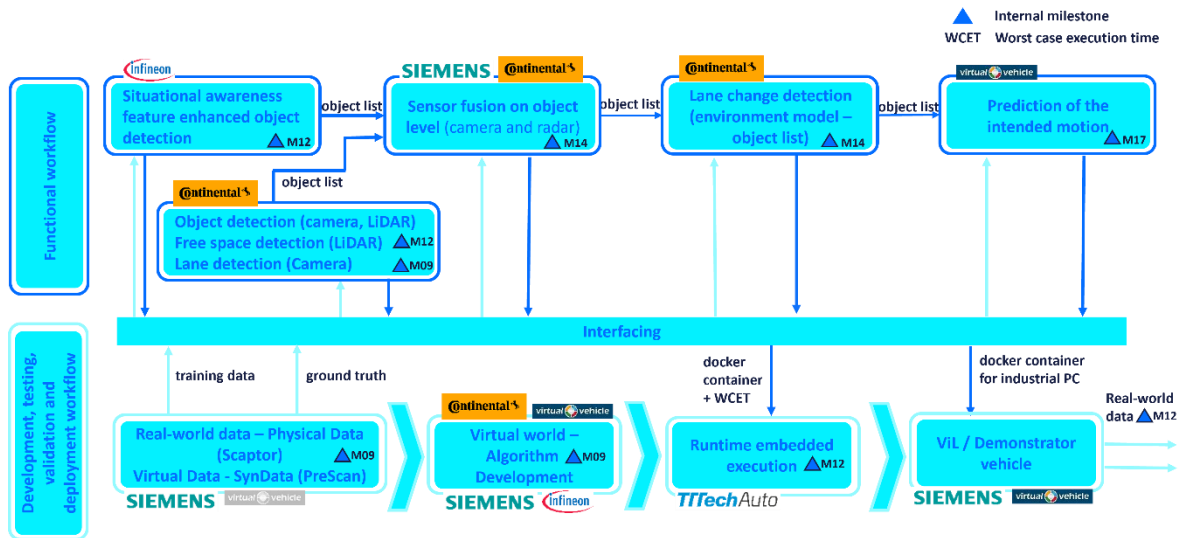


Figure 25: Use case 2.2 architecture

#### 4.6.1 Prediction of intended motion

##### Introduction

Building upon the contributions detailed in deliverable D3.2, Virtual Vehicle Research advanced its activities within UC2.2 by further developing the motion prediction model central to the robo-taxi's navigation system. Specifically, recent developments have focused on enhancing situational awareness by explicitly integrating scenario-specific knowledge and methods that may reveal the model's internal decision-making processes. These methodological enhancements have improved the model's applicability in complex urban driving contexts, characterized by their inherent diversity and uncertainties, particularly regarding the behaviour of vulnerable road users. Consequently, the chosen approach aligns closely with the overarching objectives of the AITHENA project, explicitly addressing the dimensions of explainability and robustness.

##### Methodology

The methodology adopted in the second phase of model development builds on the groundwork outlined in D3.2, advancing the motion prediction pipeline with a multi-faceted approach that integrates both inherently explainable components and post-hoc explainability techniques.

During the second phase of model development, the primary objectives were:

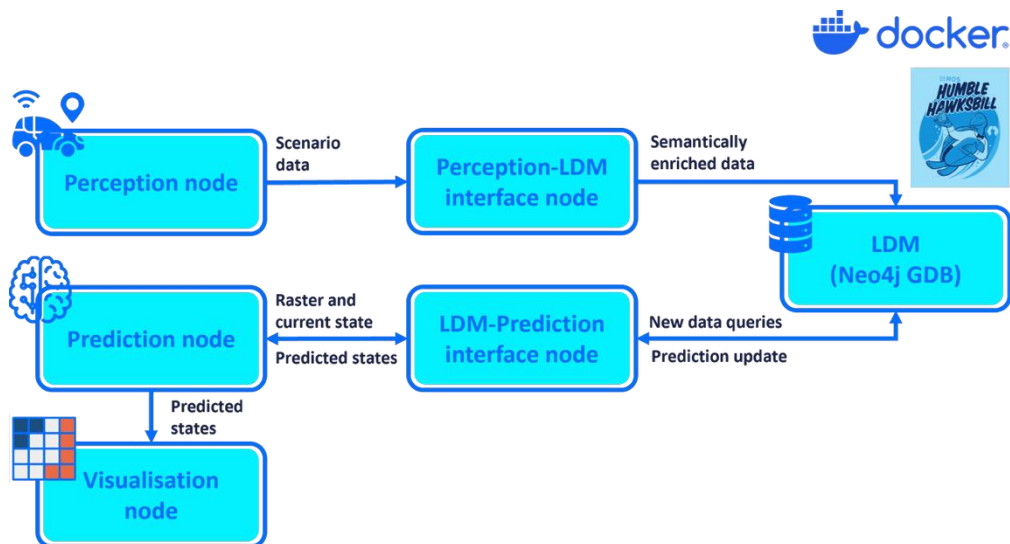
- To systematically expand the prediction horizon, assessing performance implications as predictions extend further into the future.
- To rigorously evaluate the influence of different scenario elements (e.g., infrastructure and surrounding dynamic objects) on the prediction accuracy and uncertainty for various traffic participant classes.

Central to our prediction system is the Deep Kinematic Model (DKM) [41] [42], constructed upon the efficient MobileNetV2 backbone [43]. This model incorporates explicit kinematic constraints tailored to specific classes of traffic participants, ensuring physically feasible predictions. Its design provides flexibility in dynamically adjusting the prediction horizon, directly addressing our first objective regarding the analysis of extended future predictions.

To further address the objective of evaluating scenario element influences and enhancing transparency, we integrated Grad-CAM, a gradient-weighted class activation mapping technique [44]. Grad-CAM identifies critical regions within raster inputs considered significant by the model. This integration provided clear insights into the importance attributed by the model to infrastructure features such as lanelets and crossings, and dynamic objects including vehicles and pedestrians. By visualizing these learned regions of interest, Grad-CAM directly supports the systematic evaluation of scenario elements on prediction performance.

Supporting these core elements is our ROS2-based, Dockerized pipeline (see Figure 26) which also incorporates a Local Dynamic Map (LDM) implemented using Neo4j graph database (GDB). The LDM maintains relevant infrastructure and historical scenario data within a predefined temporal horizon, enhancing the explainability of the overall pipeline by providing structured semantic and contextual information.

In conclusion, our methodological approach, combining explicit physical constraints, deep learning efficiency, and visual explanations, strongly supports the explainability and robustness objectives.



**Figure 26: Prediction generation and result visualisation pipeline**

## Results

Extensive testing was conducted to evaluate the performance of the developed prediction model, focusing on understanding the impact of varying model configurations and input data. Structured as comprehensive ablation studies, the tests systematically examined the following parameters:

**Traffic participant class:** pedestrians, cyclists, vehicles.

**Movement influences:** environmental context alone and combined with dynamic obstacles, exploring varying distance thresholds of their influence.

**Width of the prediction horizon:** varied between 1 and 6 seconds.

**Kinematic model:** tested were Constant Velocity (CV) with multiple separate cells, CV with one shared cell (weights and biases), and Constant Acceleration (CA) with one shared cell. While the architecture

supports Constant Turn Rate and Velocity (CTRV) and Constant Turn Rate and Acceleration (CTRA) models, these were not evaluated due to dataset limitations.

**Loss definition:** comparisons between considering only the final predicted state versus incorporating all predicted intermediate states. **Width of the past horizon:** evaluated using historical data windows of 50, 100, and 150 timestamps.

**Sampling period:** consistently used a default of 0.04 seconds.

**Model architecture:** performance comparisons between MobileNetV2 and planned evaluations for MobileNetV3.

The outcomes of these ablation studies provided crucial insights into how each parameter influences prediction performance across various scenario conditions and participant types. Furthermore, these evaluations laid the groundwork for an informed selection of model configurations and parameters that best balance prediction accuracy, robustness, and computational efficiency.

The testing results will also inform the planned demonstrator setup, wherein the developed prediction pipeline is anticipated to be integrated into TTTA’s middleware solution.

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 16.

**Table 16: Prediction of intended motion - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	The model incorporates physical and mechanical constraints corresponding to the specific class of the traffic participant, enhancing equitable treatment across different participant classes.	Representation accuracy for certain traffic participant classes is limited by the availability and completeness of the training datasets (e.g., participants using various mobility aids are underrepresented).
<b>Transparency</b>	Model transparency is significantly enhanced through the integration of inference-level constraints and post-hoc explanations using Grad-CAM, coupled with the provision of detailed Model Cards documenting its functionality.	The current XAI approaches, while providing meaningful visual explanations, do not fully reveal the intricate internal decision mechanisms of the deep learning components.
<b>Accountability</b>	The integrated XAI methods collectively contribute towards enhanced accountability of the predictive outcomes by clearly indicating influential scenario elements and limiting predictions to physically plausible trajectories.	The accountability provided by the model is inherently limited to the operational domain and scope defined by the training data and model architecture.
<b>Privacy</b>	Operates exclusively on data provided by the perception system (states of observed	N/A

	traffic participants) or on offline loaded infrastructure information, thus ensuring no personal data or sensitive information is processed.	
--	--	--

## 4.6.2 Situational awareness feature enhanced object detection

### Introduction

IFAG developed algorithms for object detection with situational awareness features, which are used for robust scene understanding/prediction based on camera data. These algorithms are embedded in the robust prediction modules for robo-taxis in urban environments within UC2 (AI extended Situational Awareness / Understanding).

### Methodology

The following Situational Awareness features have been addressed in detail:

- Feature 1: traffic light colour detection (red, orange, green)
- Feature 2: detection of vehicle signal lights (brake lights, left/right indicators), and
- Feature 3: bounding box orientation of detected objects.

For all three target features, algorithms were designed and developed using deep learning approaches based on the well-established one-step detection strategy "You Only Look Once" (YOLO). The algorithms were initially trained on several public datasets, such as LISA, CARLA, Kitty, Cityscapes and Eurocity.

Feature 1 and 2: Meta-YOLOv8 models (single stage detection models selected for speed detections and edge compatibility) were developed using meta-learning to enhance adaptability and precision in detecting traffic light colours and rear vehicle signals, incorporating post-processing layers for geometric reasoning and contextual robustness under real-world urban driving conditions. For vehicle signal detection (left/right/brake), a geometric post-processing layer was added to Meta-YOLOv8, leveraging spatial context (e.g., object position and symmetry) to improve classification accuracy.

Feature 3: An attention-enhanced YOLOv12 object detection pipeline was augmented with real-time camera motion compensation using optical flow and affine transformations. By stabilizing object trajectories and analysing inter-frame displacement with ID-based tracking and spatial thresholds, the system accurately infers directional movement (left/right), enhancing robustness in dynamic and cluttered driving environments.

## Results

The following paragraphs deal with the results gathered relating to the three targeted situational awareness features, with a focus on the AI algorithmic aspects.

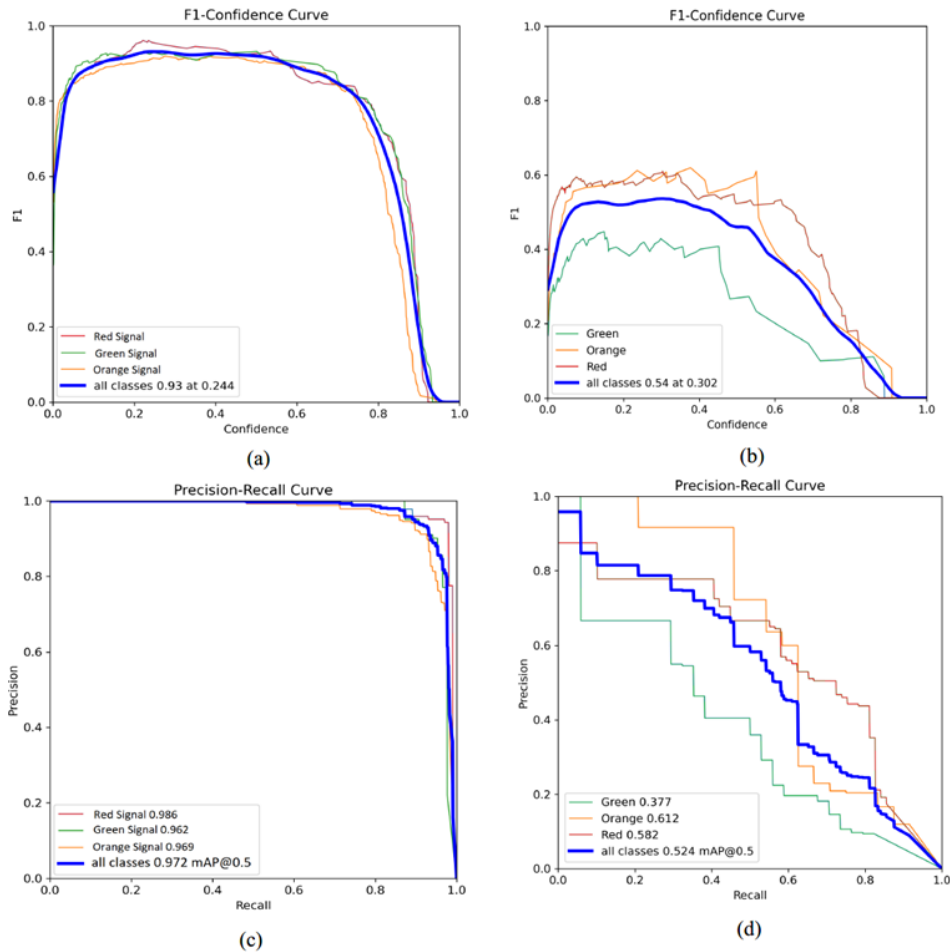
### Results feature 1: Traffic Light Colour Detection

Figure 27 (a) and (b) show the F1-Confidence curves for the proposed Meta-YOLOv8 model and a baseline model, respectively. The best-performing model (Figure 27 a) achieves a peak average F1 score of 0.93 at a confidence threshold of 0.244, indicating a well-balanced precision-recall trade-off across red, green, and orange signal classes. In contrast, the baseline model (Figure 27 b) demonstrates limited performance, with a maximum F1 score of only 0.54 at 0.302, and noticeable instability, especially in the green signal class.

Figure 27 (c) and (d) present the corresponding Precision-Recall curves. The Meta-YOLOv8 model (Figure 27 c) delivers a high mean Average Precision (mAP) of 0.972 at IoU = 0.5, with a strong precision-recall balance:

- Red: Precision 0.986, high recall
- Green: Precision 0.962, high recall
- Orange: Precision 0.969, high recall

This indicates both high confidence and effective generalization across all classes. In comparison, the baseline model (Figure 27 d) achieves a significantly lower mAP of 0.524, with degraded performance especially in the green class (Precision 0.377, lower recall), suggesting frequent false negatives and missed detections.



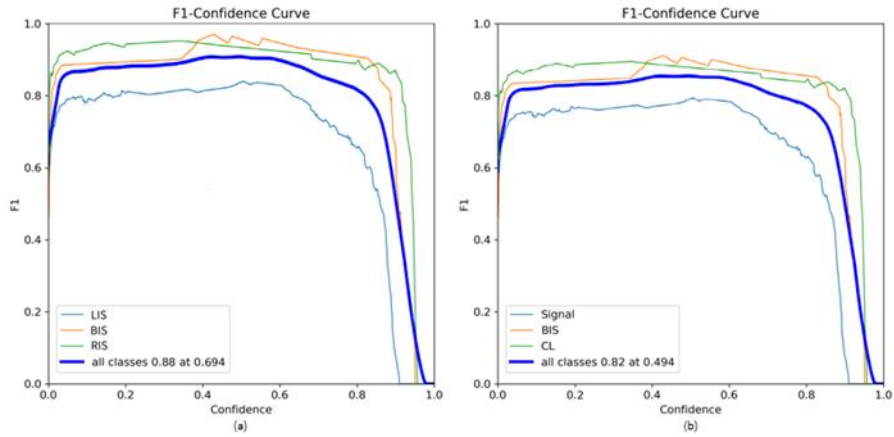
**Figure 27: Results feature 1: F1-Confidence and Precision-Recall Curves.**

### Results feature 2: Detection of vehicle signal lights (brake lights, left/right indicators)

This task focuses on detecting vehicle rear signals: Left Indicating Signal (LIS), Right Indicating Signal (RIS), Brake Indicating Signal (BIS), using a Meta-trained YOLOv8 model, both with and without a custom Post-Processing Layer (PPL).

Figure 28 (a) shows results for the best-performing model: Meta-YOLOv8 with Post-Processing Layer (PPL). It achieved a high F1 score of 0.88 at a confidence threshold of 0.694. The PPL leverages geometric reasoning to resolve spatially ambiguous signals, significantly improving classification consistency for LIS and RIS.

Figure 28 (b) represents the baseline: Meta-YOLOv8 without PPL, which achieved an F1 score of 0.82 at 0.494. This version lacked spatial correction and used base model to develop final model.



**Figure 28: Results feature 2: F1 Score Analysis.**

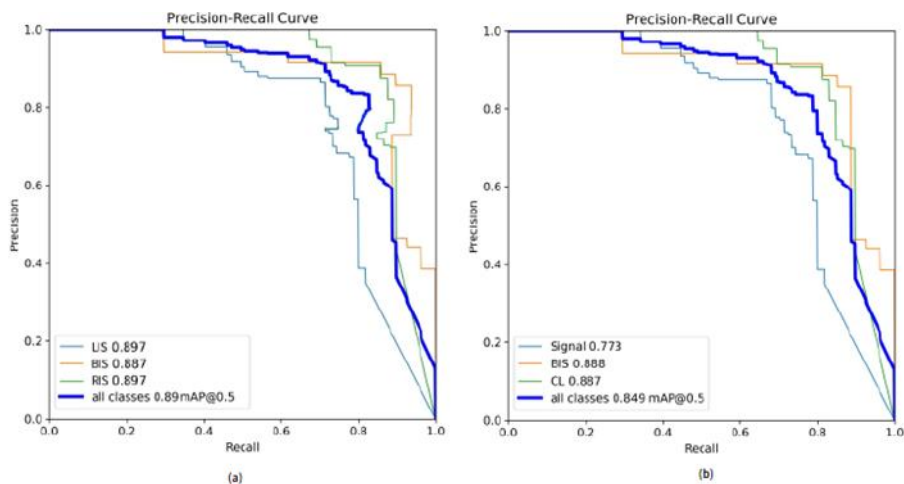
Figure 29 (a) (with PPL) reports a strong mean Average Precision (mAP@0.5) of 0.89, with class-wise precision:

- LIS: 0.897
- RIS: 0.897
- BIS: 0.887

Figure 29 (b) (without PPL) shows a reduced mAP@0.5 of 0.849, with precision values:

- Signal (LIS+RIS combined): 0.773
- BIS: 0.888
- CL (Center Light): 0.887

These results confirm that the inclusion of PPL enhances both precision and spatial resolution, while the baseline model suffers from class overlap in LIS/RIS due to lack of positional context.

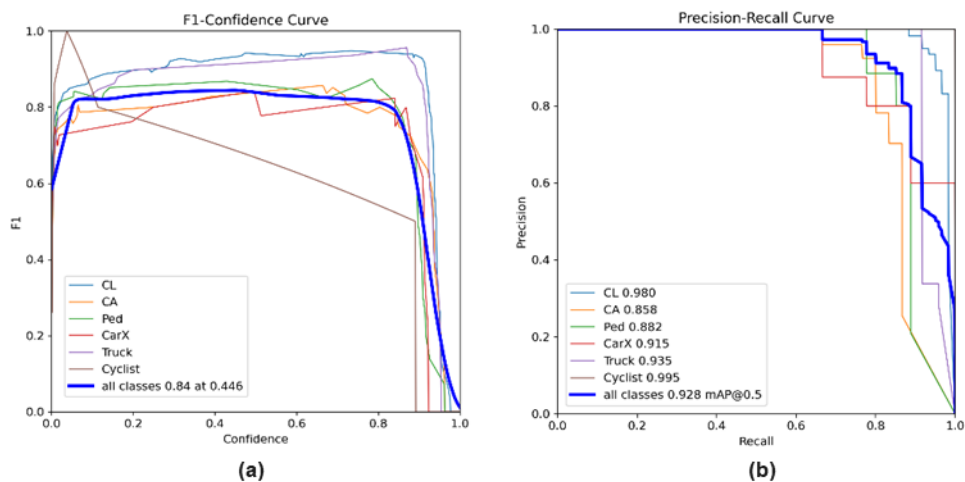


**Figure 29: Results feature 2: Precision and mAP Evaluation.**

Results feature 3: Bounding box (BBox) orientation of detected objects

The F1-Confidence curve (see Figure 30 a) shows how well the model balances precision and recall at different confidence levels. The best overall performance is reached at a confidence score of 0.446, where the average F1 score across all classes is 0.84. This means the model is most reliable at this threshold. Classes like CL (car leaving), Truck, and Cyclist perform very well and remain stable. However, CA (car arriving) and Ped (pedestrian) show more variation and drop in performance at higher confidence, which may be due to unclear signals or fewer examples in the training data.

The Precision-Recall curve (see Figure 30 b) gives more detail about how accurately the model detects objects. The average precision for all classes is 0.928, which is very good. Classes like Cyclist and CL achieve high precision and recall, meaning they are detected correctly most of the time. On the other hand, CA and Ped have lower precision when recall is high, indicating some false detections. This suggests the model could be improved for these classes with better data or training strategies.



**Figure 30: Results feature 3: F1-confidence and precision-recall curve.**

Parallel processing was implemented to reduce the bottleneck between the model inference and post-processing stages, enabling smoother and faster data flow across the pipeline.

The orientation and motion classification of dynamic road users, including cars (arriving/leaving), pedestrians, trucks, and motorcyclists—was achieved using a YOLOv12-based detection model combined with optical flow-based tracking and spatial post-processing with using a Kalman filter. The system first detects all relevant objects and BBox in each frame using a custom-trained YOLOv12 with self-attention model. To compensate for camera movement, the method estimates frame-to-frame motion using Lucas-Kanade optical flow and affine transformations, ensuring that object movement is calculated relative to a stable background. Detected objects are tracked across multiple frames using a position buffer, and their direction of motion (Left or Right) is inferred by comparing the horizontal shift of their bounding box centres over time. Special handling is implemented for “Car Arriving” (CA) detections using visual cues, while “Car Leaving” (CL) is used as a reference for spatial context. To maintain real-time performance, the pipeline includes frame resizing, position filtering, and quantization techniques to reduce bottlenecks between detection and post-processing stages. This combination enables accurate and low-latency classification of vehicle behaviours in complex, real-world scenes.

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 17.

**Table 17: Situational awareness feature enhanced object detection - Strength and weaknesses**

	<b>Strengths</b>	<b>Weaknesses</b>
<b>Fairness</b>	Model trained on diverse datasets (CARLA, KITTI, EuroCity, nuScenes) to improve generalization and reduce bias across environments and lighting conditions.	Underrepresentation of classes may lead to biased detection and reduced accuracy for these categories. Additionally, extreme weather conditions and high vehicle speeds (above 80 mph) can negatively impact detection reliability, resulting in deviations or poor-quality outputs.
<b>Transparency</b>	Clearly defined architecture (YOLOv8/YOLOv12), post-processing logic, and documented performance metrics (F1, mAP) improve technical clarity.	Complex post-processing and meta-learning methods reduce explainability for end-users or non-technical stakeholders.
<b>Accountability</b>	Modular pipeline (pre-processing, detection, tracking) supports debugging and traceability of detection stages.	Real-time failures in dynamic scenes (e.g., motion blur or occlusion) can be hard to trace without extensive logging or edge diagnostics.
<b>Privacy</b>	No personal identifiers or facial data are collected; only vehicle signal and object-level features are detected.	Use of onboard cameras may unintentionally capture identifiable data in public, raising privacy concerns if not properly anonymized.

### 4.6.3 Free space and lane detection

This development by CAF has not generated more results than those reported in D3.2. The efforts from this task have been reallocated in the development of a new Radar-Camera fusion and the development of a deliverable and reusable version of all algorithms (this one included).

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 18.

**Table 18: Free space and lane detection - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	Continental systems are tested across diverse road types (highway, urban, rural), which helps generalize free space and lane detection beyond ideal conditions.	Systems may perform poorly in areas with worn-out or missing lane markings, common in less-maintained roads or developing regions
<b>Transparency</b>	Free space and lane detection pipelines are modular (preprocessing → feature extraction → lane model fitting → output overlay), which aids transparency in development and debugging.	Output signals (e.g., curvature, yaw angle, offset from centre) are numerical and not easily interpretable by non-experts, making it hard for drivers, regulators, or external auditors to understand decisions.
<b>Accountability</b>	logging of detected lanes, free space boundaries, and vehicle pose	Lane detection failures can result from subtle visual noise (e.g., rain, glare) or fusion errors. Determining whether a failure lies in the sensor, the algorithm, or system-level design can be difficult.
<b>Privacy</b>	None	None

### 4.6.4 Object Detection

This development by CAF has not generated more results than those reported in D3.2. The efforts from this task have been reallocated in the development of a new Radar-Camera fusion and the development of a deliverable and reusable version of all algorithms (this one included).

An overview of the strengths and weaknesses of this algorithm based on the human-centric AI principles as defined in D1.1 [10] is provided Table 19.

**Table 19: Object Detection - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	large-scale, structured datasets including various traffic participants, improving coverage across different object classes.	Detection accuracy often varies by demographic and object types — e.g., children, people of colour, or people in wheelchairs may be less accurately detected
<b>Transparency</b>	Object detector outputs can be visualized in real time or in replay tools, allowing engineers	The internal decision-making of convolutional neural networks is hard to explain — why an object

	to inspect bounding boxes, classes, and confidence levels.	is missed or misclassified is often not interpretable.
<b>Accountability</b>	Continental applies ISO 26262 and SOTIF (ISO/PAS 21448) to the design of object detectors, ensuring safety checks, and documentation	If an object is missed, it's difficult to determine whether it's due to data imbalance, network generalization failure, sensor misalignment, or integration flaws.
<b>Privacy</b>	None	None

#### 4.6.5 Sensor fusion on object level

This development by CAF has not generated more results than those reported in D3.2. The efforts from this task have been reallocated in the development of a new Radar-Camera fusion and the development of a deliverable and reusable version of all algorithms (this one included).

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 20.

**Table 20: Sensor Fusion on Object Level - Strength and weaknesses**

	<b>Strengths</b>	<b>Weaknesses</b>
<b>Fairness</b>	Fusing inputs from radar, LiDAR, and cameras mitigates environmental bias (e.g., camera bias in poor lighting), helping the system perform more equitably across diverse weather and lighting conditions.	Scenarios involving non-standard vehicles, occluded VRUs, or rare objects may not be fairly handled due to their scarcity in datasets, even with multi-sensor fusion.
<b>Transparency</b>	modular architectures (detection → fusion → tracking), allowing traceability of object states and decisions.	As with most Tier-1 suppliers, deep access to fusion algorithms may be restricted due to intellectual property
<b>Accountability</b>	logs (timestamps, sensor IDs, object histories), enabling post-event analysis to identify where and why a misperception occurred.	In late fusion models, object-level outputs are the result of combined sensor interpretations, making it hard to isolate the source of an error and assign responsibility.
<b>Privacy</b>	None	None

#### 4.6.6 Virtual world – simulation

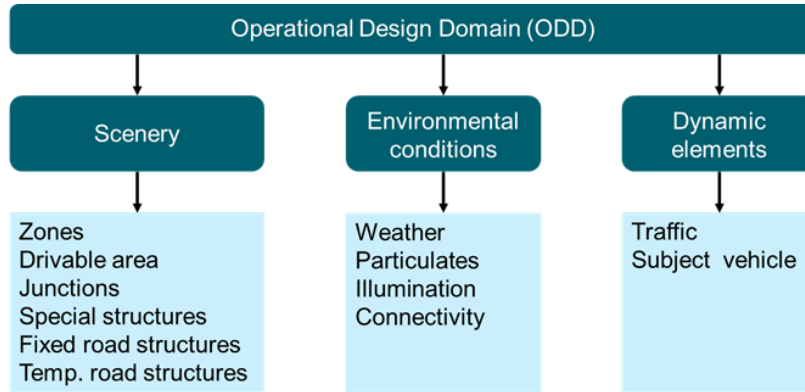
##### Introduction

Synthetic data is information that's artificially generated rather than produced by real-world events. Data generated by a computer simulation can be seen as synthetic data. This encompasses most applications of physical modelling, such as music synthesizers or flight simulators. The output of such systems approximates the real thing but is fully algorithmically generated. Synthetic data is generated to meet specific needs or certain conditions that may not be found in the original, real data.

Methodology

## Methodology

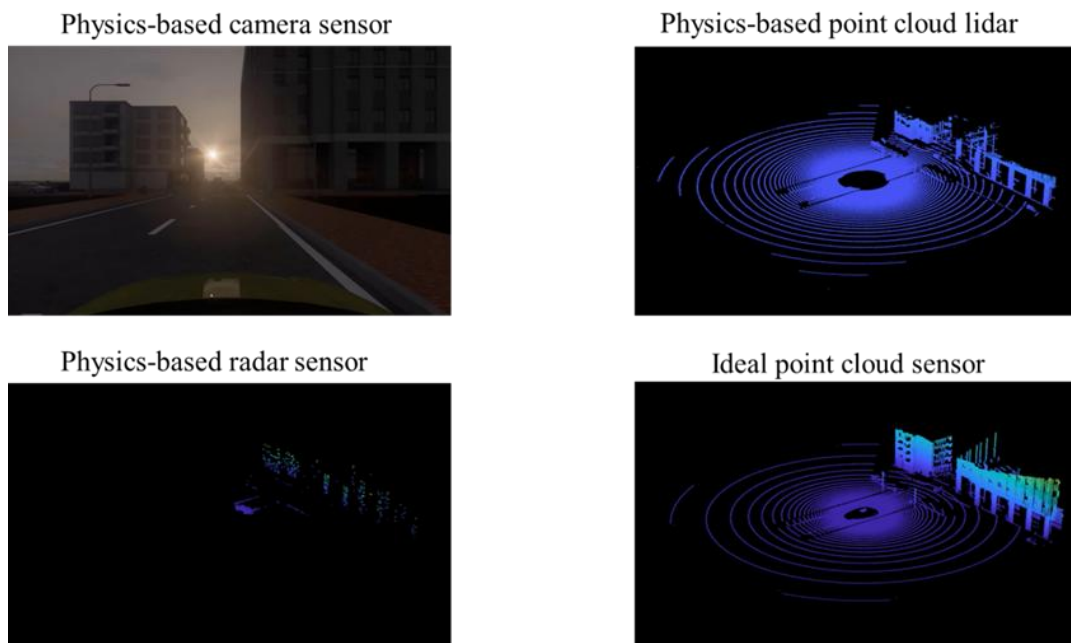
The data set is generated automatically using the simulation environment Simcenter Prescan. Since, the number of attributes of the operational design domain can be large and some attributes are continuous variables, a smart sampling approach is necessary. In this sense orthogonal arrays have been used to generate the synthetic data set.



**Figure 31: Operational Design Domain main attributes – ISO PAS 1883**

## Results

Synthetic data set generated by SIE-NL has been extended considering different illumination and weather conditions, in addition to different vulnerable road users (VRUs). Furthermore, the synthetic data contains the output of physics-based virtual sensors such as camera, LiDAR, and radar (see Figure 32).



**Figure 32: Synthetic data: camera, LiDAR, radar – generated using Simcenter Prescan**

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 21.

**Table 21: Virtual world – simulation - Strength and weaknesses**

	Strengths	Weaknesses
Fairness	Synthetic data is generated considering different safety critical events, which are not or difficult to capture in real-world	Realism, accuracy of the synthetic data. Credibility of the simulation
Transparency	Transparent approach, physics-based sensor models are used, according to well-known laws of physics.	Radar and LiDAR sensor models are complex and might be difficult to understand for people not familiar with the topic.
Accountability	Simulations are repeatable and reproduceable easily in comparison of real-world tests	Simulation credibility and validation of models using experimental data are essential and require proper engineering expertise
Privacy	No privacy related issues, since synthetic data is generated using and integrated software toolchain	N/A

#### 4.6.7 Safety-critical embedded execution concept (TTTA)

##### Introduction

TTTech Auto (TTTA) is focusing in T3.4 on developing a fault-tolerant decision-making subsystem concept as a basis for safe Automated Driving Systems. Designing the proposed architectural system concept enables SAE [45] Level 3 and 4 Automated Driving (AD) Systems for highly automated driving. It is essential to develop *safe* automated systems to prevent driving accidents. A specific focus of this module is functional safety and fail-operational performance (e.g., having all the necessary input protections needed for the automotive environment, capable of receiving trajectory data via redundant CAN buses, etc.). In the following years, tens of millions of vehicles are expected to be equipped with L4 driving systems allowing safe and reliable sensor data processing for e.g. Highway Pilot functions, Valet Parking and Autonomous truck driving in case individual functions fail. The proposed subsystem fail-operational concept ensures functional safety by design and introduces the necessary level of redundancy and diversity in a system to prevent common cause failures that are hard to predict.

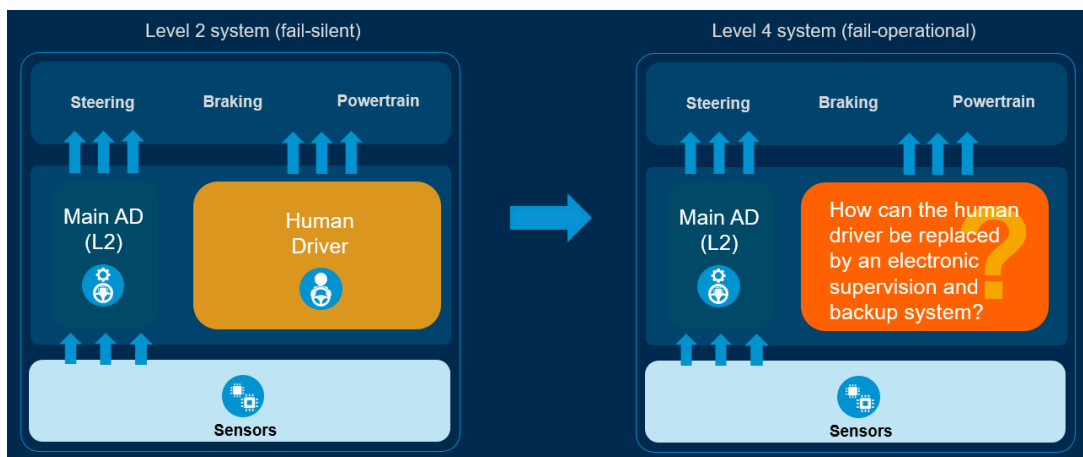
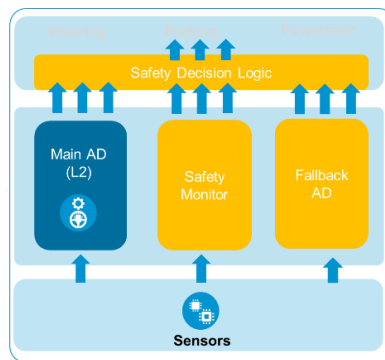


Figure 33: Move from SAE Level 2 to Level 4 system: from fail-silent to fail-operational (TTTA)

## Methodology

In the first period, we have examined the requirements and constraints that the architecture for SAE Level 4 must fulfil in real-world scenarios. It is essential that critical events, whether external or internal, as well as hardware failures, do not result in an accident. Accordingly, we have developed a concept for a fault-tolerant decision-making subsystem, detailed below in *Results*. This concept addresses the UC2.2 safety requirements and, in general, takes the high safety and reliability requirements of global car manufacturers' mass-production programs into account and has broader applicability guaranteeing fail-operational performance of the AD systems which support the driver of a vehicle in routine situations and help to reduce the number of traffic accidents. We consider functional and non-functional requirements as well as performance characteristics and quality objectives to develop this architectural concept, which will be finalized in the second reporting period of the project.



**Figure 34: SAE Level 4 system (fail-operational) mechanism (TTTA)**

In the previous project period, we focused on the further detailing the safety concept. We have specified the functionality of the Safety Monitor and the Safety Decision Logic. We worked on the designing the checks that the safety monitor would have to perform.

## Results

Figure 33 shows the differences between Level 2 and Level 4 automated driving systems. The latter supposes that the safe operation of a vehicle is a responsibility of a system and is not only a subject to a human control at a steering wheel. Today, advanced sensors and computer technologies support the driver of a vehicle in most practical situations and allow to reduce the number of traffic accidents. However, it is difficult to precisely specify all edge cases that can be encountered in driving conditions. Therefore, TTech Auto is further developing a fail-operational system mechanism that achieves a systematic partitioning into independent Fault Containment Units to manage the complexity of the L4 system. The concept introduces a Safety Decision Logic approach based on Safety Monitor and Fallback AD mechanisms, as shown in Figure 34. This decision-making subsystem concept can be used as a separate Electronic Control Unit, or ECU, which is ASIL D compliant to be used as a voter in automated driving systems.

As stated above, Safety Monitor will execute different safety checks. One relevant check is the Short-Term Collision Check (STCC) generated trajectory of the Main AD checked for potential collisions. The safety goal targeted by the STCC is that the overall system shall avoid high-severity (S2 or S3) collisions that could harm passengers (of either vehicle) or vulnerable road users (VRUs). Another check considered is Basic Feasibility Check (BFC) that should check if it will be physically difficult for the ego vehicle to follow a proposed trajectory that requires accelerations close to the (tire-road) friction limit.

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 22.

**Table 22: Safety-critical embedded execution concept - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	-	No difference made between the kind of obstacles to estimate the validity of the proposed trajectory (pedestrians, cars, or other objects) in the current concept.
<b>Transparency</b>	The concepts propose to calculate feasibility of proposed trajectory and collision risk with physical and models probability calculations which would be transparent.	-
<b>Accountability</b>	The proposed checks provide information if the planned trajectory is valid according to different properties (feasibility, safety, etc.) what makes checks accountable (e.g., for switching to fallback system).	If the checks result in a negative trajectory evaluation, the primary system is not accountable anymore. Further decisions are made by the fallback system (with limited capabilities)
<b>Privacy</b>	N/A. No personal data used or generated.	N/A

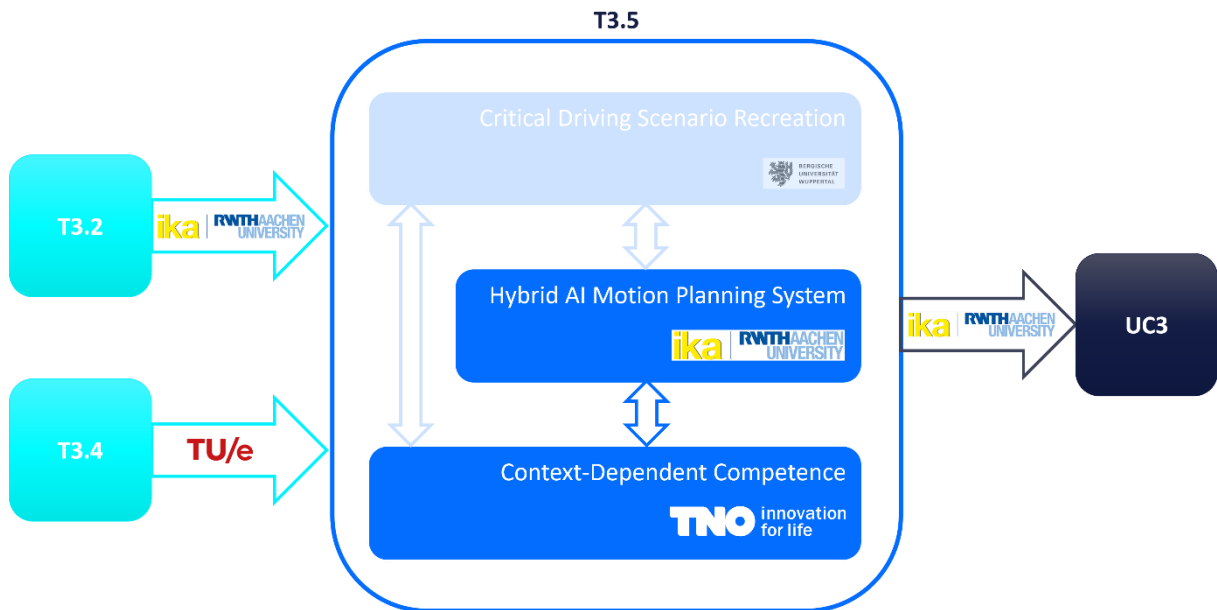
## 5 EXPLAINABLE AND ROBUST DECISION MAKING (MANOEUVRE AND TRAJECTORY)

### 5.1 Introduction

T3.5 Explainable and robust decision making is related to the superior processing step, namely the planning of reasonable driving behaviours of an automated driving software pipeline. Therefore, this chapter outlines the work on the development of AI algorithms that are related to this topic. Figure 35 provides the approaches and dependencies between the different modules that are developed in T3.5.

The implemented algorithms are based on several inputs like object lists derived from modules related to the task of perception (T3.2) and their respective predictions (T3.4). Moreover, these algorithms utilize additional sources of information like digital maps or information derived from Vehicle-to-Everything (V2X) communication. The implemented algorithms will be integrated into various demonstrators as part of UC3.

Therefore, the algorithms developed in T3.5 during the second project period are be divided into the two subtasks of the implementation of a hybrid AI motion planning system and the modelling of context-dependant competence. The subtasks are described in more detail in the following chapters.



**Figure 35: Approach and dependencies within T3.5**

The overall objective of T3.5 is to improve robustness and understandability for decision-making modules by:

- combining data-driven- and knowledge-based AI for decision-making,
- modelling the competence of data-driven AI systems for decision-making to properly react to unforeseen situations,
- explaining decisions to human occupants through visualization of decisions.

## 5.2 Hybrid AI motion planning

### 5.2.1 Introduction

The task of planning reasonable decisions and deriving the actual vehicle movement in the form of trajectories represents the final, overarching task within the pipeline of automated driving systems. This is followed by calculating the control variables for the actuators based on the generated trajectories, which then lead to the final realisation of the planned movement.

In this final processing step, all previously generated information, e.g. from the perception and prediction modules, is used and combined with other information sources such as digital maps or V2X information.

For this reason, the accuracy and robustness of all previously generated information influences the accuracy and robustness of the decisions, planned manoeuvres, and resulting trajectories. To counteract this, algorithms for motion planning should consider potential uncertainties in the underlying information and at the same time maximise their own process-related level of accuracy and robustness.

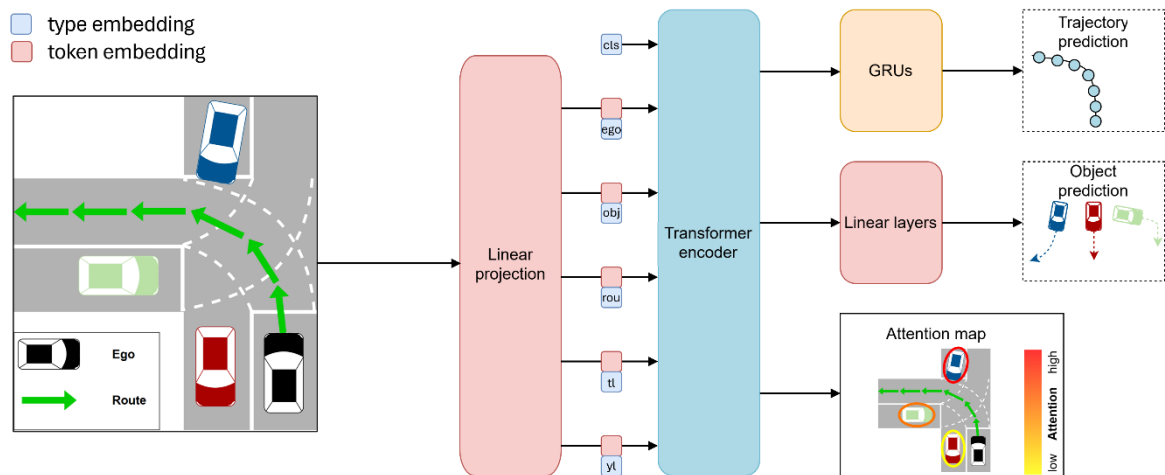
### 5.2.2 Methodology

Within this subtask of T3.5, IKA focuses on planning reasonable trajectories in a robust manner mainly by using as much information as possible from different sources of information and by combining knowledge- and data-driven AI in the motion planning system.

The implemented system consists of a sequential combination of a Deep-Learning based approach for generating an initial trajectory and a knowledge-based approach that investigates and adapts the planned trajectory based on various rules, e.g. on drivability due to driving dynamics.

The results of the algorithms developed in the first Althena Cycle have revealed a particular challenge in the integration of approaches to motion planning using AI methods, namely the performance and robustness of the vehicle guidance in a closed control loop. Approaches based on supervised learning have the problem that they are trained using an open control loop, resulting in poor performance when the control loop is closed.

One focus of development in the second phase of the project was to enable the neural networks to take the stabilisation aspects of vehicle control into account to improve closed-loop performance. To this end, a transformer-based neural network was implemented which, in addition to information on the vehicle environment, also uses explicit information on the actual own driving dynamics. This is intended to implement the concept of bi-level stabilisation according to Werling [46] in vehicle-guidance architectures including AI-based models.



**Figure 36: Implemented model architecture based on the *PlanT* [47] transformer model**

The implemented model architecture is illustrated in Figure 36. We extended the *PlanT* [47] transformer model with explicit input information on the current ego-state to be able to implement the concept of bi-level stabilisation. All inputs are encoded in a vectorized format with different information depending on the type of information (ego-vehicle, dynamic objects, route information, traffic-lights and yield-lines). All input vectors have in common that they include the information on a bounding box indicating spatial information on the specific type of objects.

The model outputs a predicted trajectory for the ego-vehicle as well as short term predictions of other objects in the scene. Lastly the model outputs attention maps representing the relative importance of the various input-tokens, enabling explainability for the model's decision.

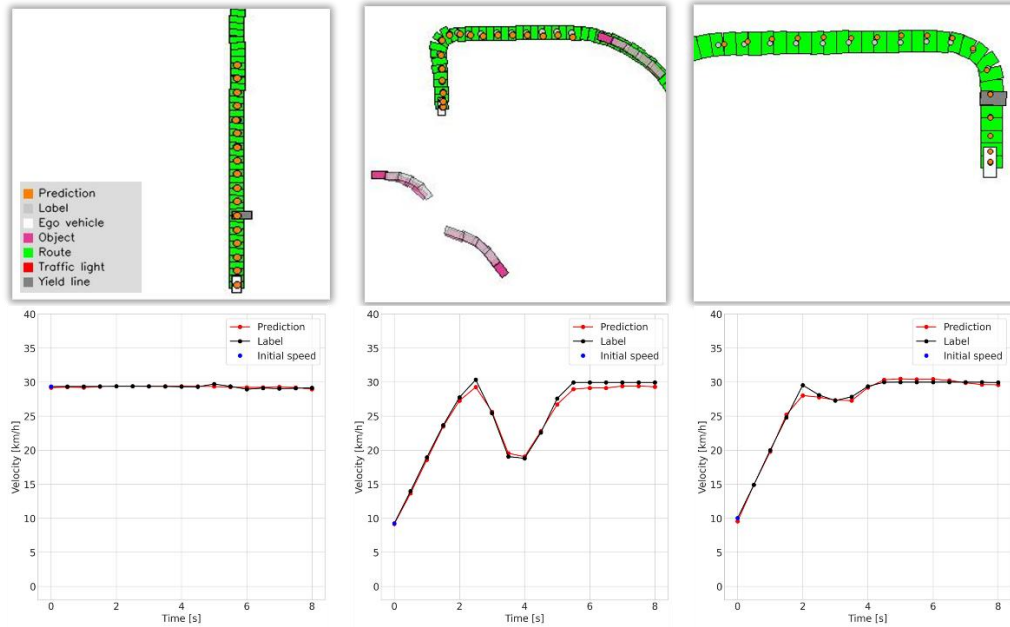
The model was trained on data of the *nuPlan* Dataset [48]. One drawback of the *nuPlan* Dataset for our application is, that it was mainly recorded in USA and Asia meaning that specific European road characteristics are not represented in the data. To account for this we captured synthetic data, based on the Digital-Twin of the *Aldenhoven Testing Center* and our research vehicle *karl*, implemented in WP4. In contrast to object detection, it is more difficult to generate the input- and label-data for training of models for behaviour planning: Good and bad driving behaviour is often very subjective. To counteract this problem, a data framework was developed to harmonise and characterise behavioural data so that it can then be classified. The aim is to be able to preselect the training data so that the neural network can be specifically trained with 'good' data. In addition, the corresponding behavioural data can be enriched with further environmental information (e.g. weather and road conditions) to account for the potential change in driving behaviour under these conditions.

Moreover, the overall architecture was refined conceptually, and the downstream rule-based algorithm was further developed.

The key findings from the Althena Cycle II are briefly explained in the following.

### 5.2.3 Results

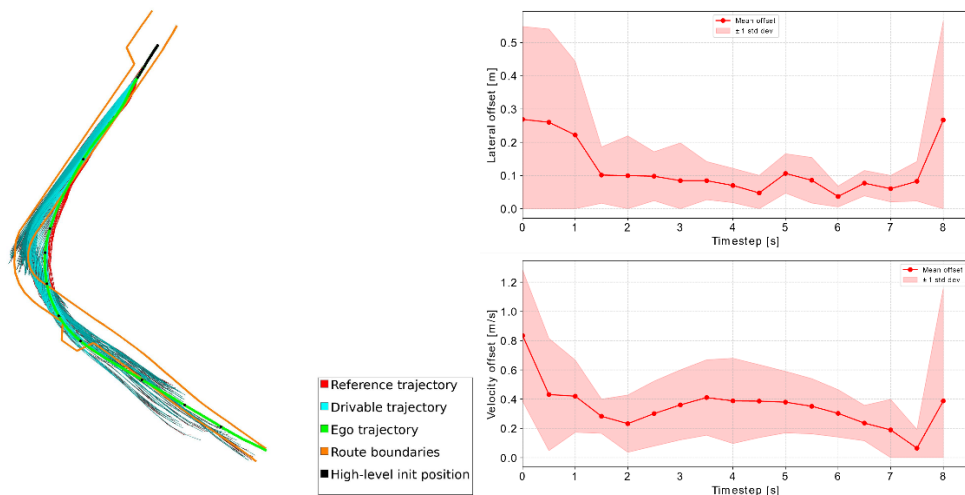
We evaluated the performance of various model variants against each other using the Mean-Absolute-Error to select the best model variant. Figure 37 shows qualitative results of the model prediction in three scenarios representing straight driving and right- and left-turn manoeuvres. The results were generated using the best model variant.



**Figure 37: Qualitative Results of best model variant in three different scenarios**

The illustration shows that the learned behaviour nearly identically matches the ground-truth behaviour. The velocity plots additionally show that especially the predicted initial vehicle speed equals the given input speed which is a necessary requirement for closed-loop stability of the system. This behaviour shows a significant improvement compared to the model proposed in D3.2.

To evaluate the closed-loop performance of the model, we integrated it into our Hybrid-AI motion planning architecture and performed closed-loop simulation using our simulation framework based on CARLA [49] (cf. D4.2 and D5.2).



**Figure 38: Results from closed-loop simulation in a left-turn scenario**

The left-hand side of Figure 38 shows a top-down view of the driven Ego trajectory alongside the predicted (Reference trajectory) and optimized (Drivable trajectory). It can be stated that the vehicle is able to follow the planned trajectories, which indicates that the efforts to improve vehicle stabilization are having an effect. This is underlined by the plots on the right-hand side of the figure, which illustrate the mean and standard deviation in terms of lateral displacement and speed.

It is noticeable that the vehicle leaves the route boundaries. However, this behaviour is intentionally caused by the model, which can be explained by the training data: the desired behaviour in the data was specified by the CARLA driver model. This driver model also intersects the specified route boundaries. In conclusion, it can be stated that it is possible to learn desired behaviour and execute it using the implemented Hybrid-AI architecture. In this case the downstream trajectory-optimization module did not consider the route-boundaries as well which led to these being passed. With consideration of the boundaries in the downstream safety module, such misbehaviour can be intercepted. Additionally with accurate training data, the model performance can be improved in the future so that the predicted behaviour also corresponds to the desired target behaviour.

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 23.

**Table 23: Hybrid AI Motion Planning - Strength and weaknesses**

	<b>Strengths</b>	<b>Weaknesses</b>
<b>Fairness</b>	By providing individual behaviour data, the resulting driving behaviour can be aligned to personal preferences of the users.	Learned behaviour is highly dependent on the given training data. A large amount of data is required to specify a generally accepted target behaviour.
<b>Transparency</b>	Explainability of the deep learning-based model can be increased through the provided attention maps indicating the importance of various inputs for the model's decision. Additionally in combination with the downstream, rule-based trajectory optimization, the resulting driving behaviour should be transparent, as the safeguarding rules are transparent.	Still does not give an explicit output why a specific decision has been predicted.
<b>Accountability</b>	By combining AI-models with a sequential rule-based algorithm for trajectory-optimization, the predicted trajectory is evaluated online. In the event of unexpected behaviour, appropriate safety manoeuvres can be initiated.	None
<b>Privacy</b>	No GDPR related data used for training of the model.	None

### 5.3 Generating trajectories for critical driving scenarios

This development by BUW has not generated more results than those reported in D3.2. The efforts from this task have been reallocated towards T3.2 instead (see Sections 3.2 and 3.5).

### 5.4 Situation awareness: Competence, context, and risk

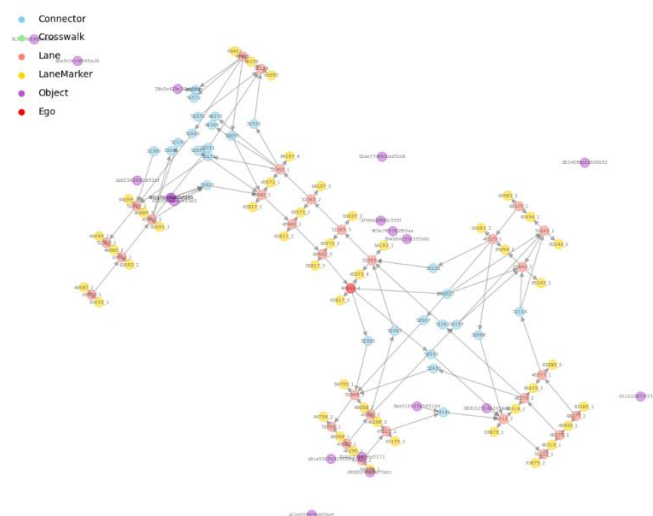
#### 5.4.1 Introduction

A major challenge for trustworthy AI lies in the gap between data-driven algorithms (in this case the trajectory planner) operating on a sub-symbolic (statistical) level, and their requirements and Operational Design Domain (ODD), which are defined on the symbolic level. While performance on test data reflects statistical trustworthiness, it doesn't ensure the system meets symbolic-level requirements (e.g., navigating urban intersections in Germany). To address this, TNO developed a situation awareness module to assess AI trustworthiness symbolically.

A key part of situation awareness is context-dependent competence—whether the system can meet its requirements in each context. Competence depends on prior experience and context complexity; familiar, simpler situations yield higher competence. Low competence doesn't imply failure, but indicates reduced reliability compared to better-known scenarios.

#### 5.4.2 Methodology

To formalise the context, we introduce the concept of a 'scene' which describes the current surroundings of the automated vehicle. We use a knowledge graph database to model the scene on the symbolic level. This type of database is designed to model knowledge using a graph structure, which contains nodes, edges, attributes, and labels, and focuses on modelling the relationships between entities. An example can be seen in Figure 39. Entities can be represented in a graph using nodes. Each node is equipped with a collection of labels and attributes to distinguish it from other nodes within the graph. Relationships between nodes are modelled using edges, with each edge connecting a pair of vertices. Like nodes, edges can also be distinguished using labels and attributes.



**Figure 39: Example of a knowledge graph representing the context.**

We have encoded the dataset that the trajectory planner has been trained on into the knowledge graph, using an ontology of urban driving. This way, we elevate the sub-symbolic description of the ODD to the symbolic level. This has several advantages:

- Competence estimation: we can define a measure of competence of the trajectory planner given the current context.
- Risk assessment: we can reason about the risk of the actions of the vehicle given the competence and the context.
- Providing explanation: we can ask the system for explanations about competence, context, and risk on the symbolic level.

For this deliverable, we focus on the first element, competence estimation. The graph representations of scenes encountered in the dataset are queried for specific sub-scenes, which encode parts of the context or the vehicle. While the entirety of the graph represents the scene in which the ego is finding itself in, sub-scenes are specific patterns within the graph structure. In total, we query for 8 separate sub-scene patterns: ego driving on straight road, ego driving in roundabout, ego enters roundabout, ego leaves roundabout, ego crosses intersection, ego approaches intersection, ego approaches pedestrian crossing, vehicle driving ahead of ego. More of these relevant patterns could be added.

When we query for all the scenes in the dataset, the total coverage of these sub-scenes in the dataset can be calculated with the same method as done in [50]. As the ego vehicle finds itself in a (new) situation, the context can be modelled as a knowledge graph and the coverage of the occurring sub-scenes can be calculated for the training set. Besides the coverage of the scene in the training, a complexity measure is also used to estimate the competence of the trajectory planner following [51]. The coverage and the complexity of a scene are combined according to the following equation:

$$Competence(s) = Coverage(s) \cdot (1 - Complexity(s))$$

### 5.4.3 Results

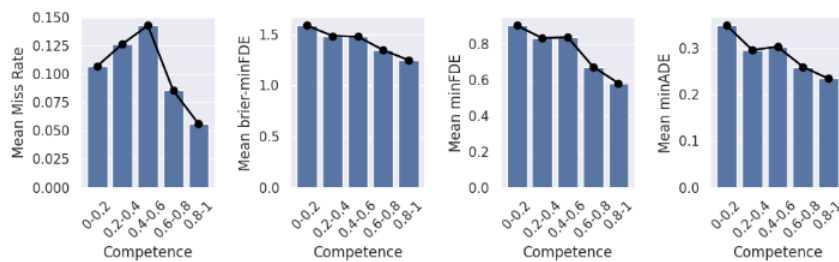
We demonstrate how the competence measure can be used to assess the competence of a trajectory planner. We show results for a DNN trajectory planner with the Autobot architecture [52] that was trained on Singapore data from the NuPlan [48] dataset. The Boston data from the NuPlan dataset was used for the evaluation of the competence metric and the comparison to the performance of the model. The training of the Autobot trajectory planner was done using the UniTraj [53] framework.

**Table 24: Correlation between competence metric and performance metrics of the trajectory planner**

	correlation	p-value
minADE	-0.092	0.0035
minFDE	-0.087	0.0059
brier-minFDE	-0.090	0.0046
Miss Rate	-0.103	0.0012

To assess the competence metric, the model was evaluated on the Boston data. The following four common evaluation metrics for trajectory prediction were used. The Minimum Average Displacement Error (minADE) is the mean Euclidean distance between the points of the ground truth trajectory and

the predicted trajectory. The Minimum Final Displacement Error (minFDE) is like the minADE, but only the final point in the trajectory is considered. The Brier Minimum Final Displacement Error (brier-minFDE) is the minFDE, but it considers the prediction confidence of the trajectory based on the Brier score. The Miss Rate is defined as the ratio of trajectories for which the minFDE is greater than 2 meters. Additionally, the competence metric was calculated for these scenes. The Pearson correlation was calculated between the competence and these evaluation metrics. We assume there is a relationship between them because the trajectories should be accurate when the competence is high, but unpredictable when the competence is low. Table 24 shows that there is a weak but significant negative correlation between the evaluation metric and the competence metric. Figure B shows the mean value of the evaluation metrics over the competence and Figure 40 shows four trajectory examples with the calculated competence.



**Figure 40: Mean value of the evaluation metrics over the competence metric binned in steps of 0.2.**

These results show that overall, there is a connection between the competence metric and the actual performance of the trajectory planner. However, the fact that it is only a weak correlation and that the Miss Rate increases for lower-level competencies can be caused by multiple factors. First, low competence means that the situation is unseen and possibly complex. This implies that the output of the model is unpredictable and should not be trusted. However, it can still be the case that the model infers a correct trajectory. Second, parts of the context that influence the performance in these cases might not be modelled in the KG or in the sub-scenes. To alleviate this problem, more context can be modelled, and additional important sub-scenes can be added.

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 25.

**Table 25: Situation Awareness: competence, context and risk - Strength and weaknesses**

	Strengths	Weaknesses
<b>Fairness</b>	By looking at the coverage of the subscenes in the training dataset, biases that are within the training data are more easily exposed.	The manual definition of subscenes gives a possibility to have a bias towards specific context compared others. This can lead to an unfair coverage metric.
<b>Transparency</b>	The knowledge graph representation and subscenes are defined on a human understandable symbolic level increasing the transparency.	No Interface has been developed to relate the competence metric to the current knowledge graph representation, to show what impacts it the most. This would increase transparency even more.

<b>Accountability</b>	The competence metric gives insight in the causes of low competence based on the breakdown between coverage of subscenes and the complexity of the scene.	-
<b>Privacy</b>	The data in the knowledge graph database does not include any privacy sensitive data.	The training data needs to be available to calculate the coverage of subscenes of the train data, possibly leading to some data sharing issues.

## 5.5 Network level impact of introducing AVs

### 5.5.1 Introduction

The previous Use Cases 1, 2 and 3 demonstrate the prediction, understanding and decision-making models needed for the autonomous vehicle to navigate the driving environment at the individual vehicle level. Use Case 4 “AI-based Traffic Management (TM)” focuses on the impact of AI-based Automated Vehicles (AVs) on the network level from a traffic management perspective.

A virtual microsimulation is used to study how introducing AVs into a typical urban traffic network influences overall traffic network dynamics and network performance. This allows capabilities of the AI models to be analysed in scenarios that includes the interaction with mixed traffic and other AI and non-AI systems operating in a real traffic network.

The traffic management simulation for this use case is intended to be an explainable, transparent demonstration of a test methodology that can be further scaled or applied to other networks. The traffic simulation includes a typical small-scale urban network with a major provincial roadway, multiple signalized and unsignalized intersections, multiple traffic scenarios with autonomous vehicles, varying levels of traffic demand and an edge case/incident. This simulation is suitable to test the methodology for a small-scale urban network and to provide explainable, predictable results based on the different scenarios.

The results will indicate the effects on efficiency, safety, and sustainability for a typical urban network and help to inform road operators and transport authorities how to manage the future transport network with autonomous vehicles.

### 5.5.2 Methodology

#### Microsimulation in PTV VISSIM

Microsimulation is employed to simulate traffic dynamics within a macroscopic network. A review of available microsimulation software tools was conducted to identify a ready-to-use tool which allows for the evaluation of macroscopic traffic scenarios. The most suitable tools are VISSIM and SUMO, which both have their advantages and disadvantages. PTV VISSIM [54] is selected for the study for its ability to simulate macroscopic traffic networks, its built-in and configurable autonomous vehicles models and flexibility of the COM programming interface to simulate edge case scenarios. The basic model assumptions are specified and applied to all simulations as described in Deliverable 3.2.

#### Simulation of Autonomous Vehicle Driving Behaviour

In PTV Vissim, traffic dynamics are calculated by iterating through small timesteps (usually 10 per second) and calculating the behaviour/action for each vehicle in the network for each timestep. VISSIM contains several sub-models, such as a car-following model, lane-changing model, lateral behaviour model, etc., to simulate complex and heterogeneous driving behaviours. The driving behaviour models provide a driving logic that governs the vehicles car-following behaviour, lane changing behaviour and other decision-making rules of the vehicle in a traffic environment. These different behavioural models each contain its own parameters and can be used to emulate various types of driving behaviours of an autonomous vehicle.

Prior to 2025, the 2020 EU Horizon CoEXist Project's autonomous vehicle model was built-into PTV Vissim. The CoExist autonomous vehicle model adapted the parameters of the Wiedemann 99 Car-Following model to reflect a new, more stochastic autonomous vehicle model. The Wiedemann 74 model was suitable for intersections, roundabouts, and arterials with speed limits less than 80 km/h; and the Wiedemann 99 model was best suited for highways and speed limits of 80 km/h and greater. The CoEXist Wiedemann 99 model for AV driving includes four variations, beginning from most cautious to most advanced:

- Rail Safe - brick wall stop distance
- Cautious - Respects the rules of the road and always adopts safe behaviours
- Normal - Very similar to non-automated vehicles with the addition of measuring and maintaining distances and speeds of the surrounding vehicles due to the onboard sensors (i.e. ACC)
- All-Knowing - Assumes perfect perception and prediction, maintains smaller gaps for all manoeuvres and situations

Since PTV Vissim 2025, the CoEXist AV driving logic based on human-driving Wiedemann car-following model has been replaced with a built-in car following model based on Adaptive Cruise Control (ACC) and Automatic Lane Change (ALC).

The ACC and ALC models are not categorized by cautiousness, such as in the CoEXist model. However, extensive previous research has shown that differing levels of AV aggressiveness plays a significant role in the traffic performance. Therefore, as part of this traffic management simulation, the parameters of the new ACC and ALC models have been adapted into 3 categories of autonomous driving behaviour - cautious, normal and aggressive driving behaviour variations. This is intended to improve on the ACC / ALC model while incorporating the behaviour variations from the previous CoExist model.

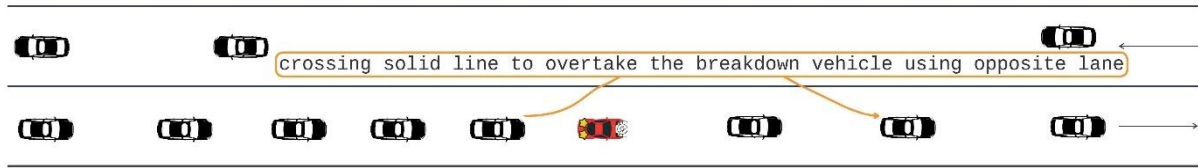
## **Simulation of Edge Case Scenarios**

In the COM programmatic interface of PTV Vissim, it is possible to create edge cases and/or specific behaviours. This interface enables the implementation of specific scenarios that are especially challenging for AVs from a traffic management perspective. A Python script implementing the incident edge case as depicted below, was created to perform each simulation as designed using the VISSIM COM interface (see D3.2 [1]). The simulation results are saved at runtime.

## **Simulation of a Proof of Concept for a Traffic Incident**

The first step was to test a proof of concept for a simple traffic incident involving autonomous vehicles on a 2-lane arterial that could be scaled up to a larger network. This proof of concept is described in Deliverable 3.2 and the results are summarized in Deliverable 5.2.

In this incident, the roadside infrastructure (RSI) detects a blocking vehicle (construction / roadworks vehicle or a breakdown) on the lane in one direction and broadcasts messages to all vehicles on the network. The single-lane, two-direction road is separated with solid line. Normal road traffic rules do not allow for the vehicles to cross a solid line, but the vehicles following the incident must overtake the breakdown vehicle and temporarily cross into the opposite lane when a suitable gap is available to continue traveling on the lane (see Figure 41).



**Figure 41: Schematic design of a vehicle breakdown incident on single-lane urban road**

The traffic mix include non-AVs, AV-with-trust and AV-without-trust. Trust is used as a proxy for how well the AV can react on unexpected incidents, external guidance and other edge cases which might challenge the AI capability of the vehicle. If an AV is an AV-with-trust, it overtakes using the opposite direction lane as advised by the messages; If an AV is an AV-without-trust, it stops for 10 seconds until its backup system (e.g., handover, remote control, remote guidance, etc.) performs the overtaking manoeuvre.

The AV behaviour is modelled using the CoEXist cautious AV driving model. One simulation run lasts two hours with the first half hour warming up and the last half hour cooling down. At 45 minutes, a vehicle breakdown incident occurs on the eastbound link. The incident lasts for 590 seconds, during which the breakdown vehicle is stopped and blocks the eastbound single-lane urban road. The following vehicles react as indicated in the UC4 algorithm flowchart of Deliverable 3.2. Just before the 55-minute timestamp, the breakdown vehicle is moving on, and the traffic returns to normal.

Three different dimensions are defined to perform simulations: percentage of AV among all vehicles, percentage of AV-with-trust among AVs, and Level of Services (LOS) [55]. In the first cycle, the LOS was fixed to LOS A<sup>1</sup>. The other two dimensions are varied to define the AV composition matrix. The composition of Light Good Vehicles (LGV) and Heavy Good Vehicles (HGV) is fixed to 5% and 2% respectively in the first cycle. The table presents the AV composition matrix with each combination of percentage of AV among all vehicles (incrementing interval of 25%) and percentage of AV-with-trust among AVs (incrementing interval of 25%). With this table, 25 vehicle mixes are defined and configured in the proof-of-concept simulation network in VISSIM. Additional (in-between) mixes were used to better understand the sensitivity of the outcome given this initial vehicle mix.

**Table 26: AV traffic mix for traffic incident (with and without trust)**

Percentage of AV among all vehicles	Percentage of AV-with-trust among Avs					
	0%	25%	50%	75%	100%	
0%	0-0	0-0	0-0	0-0	0-0	
25%	0-23	6-17	12-12	17-6	23-0	
50%	0-47	12-35	23-23	35-12	47-0	
75%	0-70	17-52	35-35	52-17	70-0	
100%	0-93	23-70	47-47	70-23	93-0	

To account for stochastic variations, each simulation of the same dimension (i.e., a fixed vehicle mix with a certain percentage of AVs and a certain percentage of AV-with-trust under a fixed LOS) is run 10

<sup>1</sup> To achieve a functional proof-of-concept and exclude server congestion, i.e. congestion upstream propagation creating gridlock, LOS A is calibrated to 1200 vehicles per hour on link 1 (incident link) and 300 vehicles per hour on link 2 (opposite direction link where vehicles overtaking break-down vehicle manoeuvres take place).

times, while the seed is increased by 78 with each iteration starting at 42 and scripting was created to automatically process and calculate the results.

The results of the simulation total vehicle delays are explained in Deliverable D5.2 [56]. There was an increase in total vehicle delay on the network in the simulations where there was a higher percentage of AVs without trust in the traffic information compared to both non-AVs and AVs with trust in the traffic information that can react more quickly on the incident. The non-AVs stop behind the breakdown vehicle, observe the oncoming traffic, and take over by crossing the solid line (across and back) safely. The AVs without trust are shown to stop behind the breakdown vehicle and show undesired cautious behaviours due to the challenging scenario. The proof of concept proved the expected effect of introducing autonomous vehicles in the network.

With this verification of the results, the simple traffic incident could be added to a larger traffic management use case which represents a typical urban network.

## Simulation of Traffic Management Use Case for an Urban Traffic Network

### *Overview of the Network*

The Traffic Management Use Case scales up the simple traffic scenario and creates a typical urban network with multiple scenarios and parameters.

The use case simulates an urban traffic network in which an SAE Level 4 autonomous driving vehicle must operate in mixed traffic with human driven vehicles and other road users. The simulation is modelled on the city of Heemstede, Netherlands. The simulation focuses on a typical 2-lane provincial road segment (N208) with 1 northbound and 1 southbound lane and several connector roads forming a small-scale traffic network.



**Figure 42: Typical intersection of the Heemstede network**

The network includes both signalized and unsignalized intersections. The signalized intersections are programmed to run according to vehicle-actuated control, so that the cycle times of the intersections change dynamically depending on the traffic demand. The unsignalized intersections are programmed to reflect different driving behaviour depending on human-driven vehicles or autonomous vehicles. Autonomous vehicles are modelled to have slower speeds at the approach of the intersection, to reflect their cautiousness compared to human-driven vehicles.

Intersection data was used as a starting point, supplemented with real-world datasets from road operator databases, including live traffic systems data and non-AV data. A two-hour simulation period was created to reflect realistic traffic conditions, consisting of a 30-minute warm-up phase, a one-hour full demand period, and a 30-minute cool-down phase. These datasets were prepared and configured to accurately replicate observed traffic patterns within the simulation environment.

### Model Parameters

The traffic simulation model varies multiple parameters to demonstrate a typical real-world, urban network. Each parameter has several dimensions including:

- Traffic Scenarios (AV and non-AV Mixed Traffic)
- Level of service (A, B or C)
- Autonomous Vehicle Driving Behaviour (cautious, normal, aggressive)
- Traffic Incident Edge Case (AVs with trust, AVs without trust)

### Traffic Scenarios (AV and non-AV Mixed Traffic)

Each traffic scenario includes a different market penetration rate of AVs.

Traffic Scenario	Percentage of AVs (%)
Baseline (No AVs)	0%
Mixed Traffic with AVs	0, 5, 10, 15, 30, 50, 70, 100%
Mixed Traffic with AVs and breakdown incident	0, 5, 10, 15, 30, 50, 70, 100%

### Level of Service (LOS)

Level of Service (LOS) is derived from the total intersection demand and scaled up for the network. The demand is then adjusted according to a percentage of deterioration in the capacity of total vehicles per the lane.

LOS	Category	Description
A	Free Flow	Excellent operating conditions. Vehicles move at or above the speed limit with complete freedom to manoeuvre.
B	Reasonably Free Flow	Slight decline in freedom of movement, but still very good conditions. Drivers rarely feel restricted.
C	Stable Flow	Traffic flow remains stable, but speed and manoeuvrability is more noticeably affected by other vehicles.

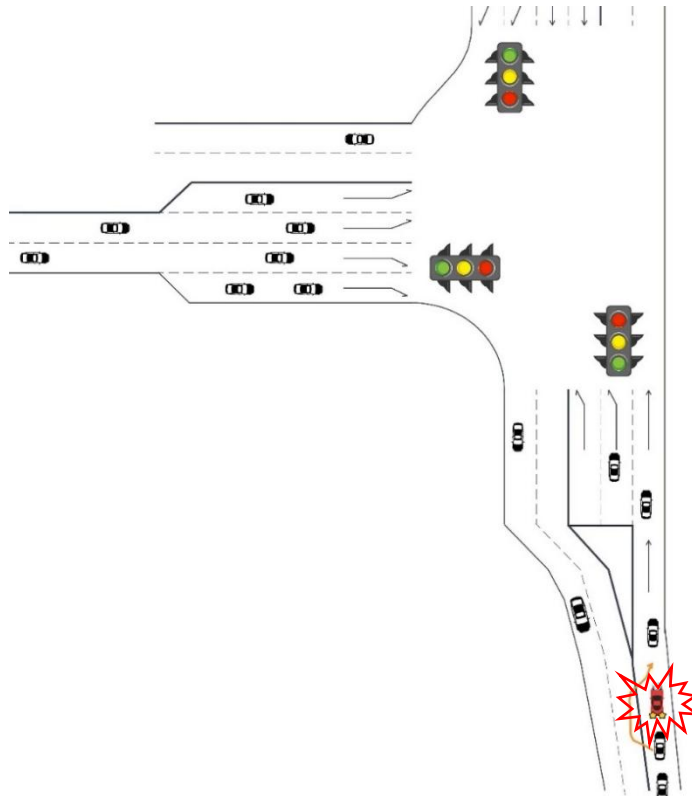
### Autonomous Vehicle Driving Behaviour

The new ACC and ALC models from PTV Vissim 2025 (see above) are more suitably designed to reflect AV driving logic because it is based on an autonomous model instead of a human driving behavioural model, such as with the CoExist autonomous vehicle model. However, the CoExist solution provides insight into the variation in AV behaviour with some vehicles designed to be more aggressive or cautious than others. To capture this variation, several parameters of the ACC and ALC model have been selected to be varied to approximate the previous CoExist models cautious, normal and aggressive behaviour. These parameters fall into the categories: car following, driving logic, necessary lane change, lane change, and signal control.

### Traffic Incident Edge Case

In the traffic management simulation for the larger network, the proof of concept for the traffic incident with AVs with and without trust is incorporated from Deliverable 3.2 as an edge case / incident in the simulation. Compared to the proof-of-concept network, the traffic impacts of the edge case / incident can now be scaled up to the larger network. The proof of concept is simplified to include only 0%, 50% and 100% of vehicles with and without trust in the external traffic guidance.

- AVs with Trust (0%, 50%, 100%)
- No AV Trust (0%, 50%, 100%)



**Figure 43: Concept of traffic incident edge case in traffic management simulation**

### 5.5.3 Results

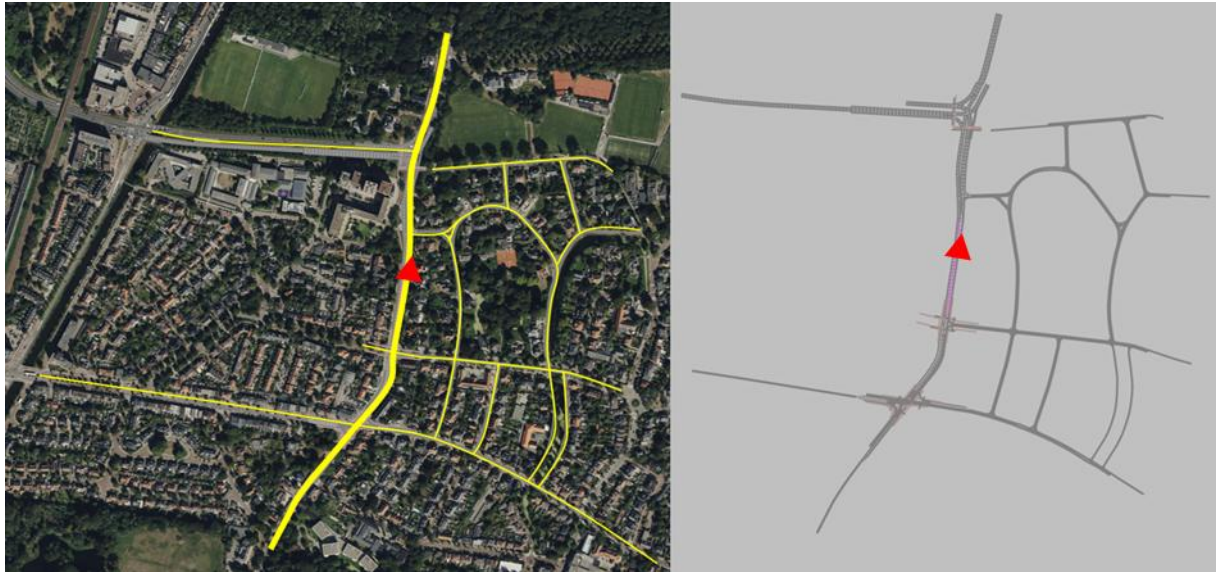
#### UC4 design – Traffic Management Simulation (Heemstede Network)

The result of this Use Case is the Traffic Management Simulation for the Heemstede network with the required logic and parameterisations.

The Vissim network models a representative urban area in the City of Heemstede, with a primary focus on local and collector roads operating at 50 km/h and 30 km/h. These roads form the core of the network, serving through traffic as well as providing access to residential neighbourhoods, schools, and retail areas. A small section of multi-lane arterial road with a 70 km/h speed limit is in the northern part of the network but plays a minor role in the overall traffic flow. The network includes three signalised intersections where the interaction between motor vehicles and vulnerable road users—such as cyclists and pedestrians—is explicitly modelled. These intersections reflect common urban challenges related to multimodal traffic and safety.

This network was developed to study the potential impact of self-driving vehicles on urban traffic. In particular, the simulation introduces an incident scenario in which several autonomous vehicles require support and become temporarily immobilized (location and direction indicated by the red triangle in the figure below). The aim is to analyse how such a scenario affects total network performance, including congestion levels, delays, and vehicle rerouting behaviour. By modelling the scenario, the network helps to explore the resilience of urban traffic systems and inform strategies for managing incidents when connected and autonomous vehicles are involved.

The results will be discussed in the upcoming Deliverable 5.3.



**Figure 44: Traffic Management Simulation (Heemstede Urban Network) in PTV Vissim**

Description of Network in VISSIM	Settings	Units
Number of links	81	
Number of connectors	110	
Total length	10,1	km
Number of controlled intersections	3	
Number of uncontrolled intersections	13	
Speed limits	70, 50 and 30	km/h

**Table 45: Description of the Traffic Management Simulation (Heemstede Urban Network)**

An overview of strengths and weaknesses of this algorithm based on the human centric AI principles as defined in D1.1 [3] is provided in Table 27.

**Table 27: Network Level Impact of Introducing AVs - Strength and weaknesses**

	Strengths	Weaknesses
Fairness	Traffic simulation does not involve any fairness or ethical issues	Traffic simulation makes assumptions of rational behaviour of AVs and non-AVs
Transparency	Provides a clear, transparent traffic simulation model depicting expected AV driving	Traffic simulation model is based on parameters of an ACC / ALC driving logic and assumptions of expected traffic conditions and

	behaviour and the impact on the network level traffic	not directly integrated with an AV driving simulator
<b>Accountability</b>	Allows for a clear evaluation of expected results using KPIs	Expected results are based on assumptions of AV driving behaviour and may differ depending on type of vehicle, manufacturer and level of connectivity
<b>Privacy</b>	Not applicable	Not applicable

## 6 CONCLUSIONS

D3.3 provides final overview of the AI algorithms developed in Althena, with a focus on the 3 main areas as defined in the Althena architecture in Figure 1: perception, situation awareness / understanding and decision-making.

Initially in D3.2 we also described more extensively the model cards, as part of the ML framework (as earlier described in D3.1 [2]), that were developed in T3.1 are a continuous development. For this deliverable however, we chose not to add any new developments on that front but rather show how these could be used in the development cycle, with 2 examples by IKA and VIF.

For each of the algorithms, background (introduction), a methodology and results are given. Additionally, we provide also an overview of advantages and disadvantages for each algorithm, considering the human centric AI principles.

Since this deliverable focused mainly on the development and explanation of the methodology used in the development of the algorithms, per algorithm results are again here described. For more detailed results and evaluation linked also to the use cases, a more detailed evaluation is provided in deliverable D5.3. One exception in this case, is use case UC2.2; since this use case is strongly linked to all four main development tasks of WP3 (T3.2 – T3.5) with many partners collaborating and it has already been in full development since the early beginnings of the project.

## 7 BIBLIOGRAPHY

- [1] J. den Ouden *et al.*, “Althena D3.2 - Report on initial AI algorithm development.” Oct. 31, 2023.
- [2] P. Cañas *et al.*, “Althena D3.1 - Life cycle management framework for machine learning models.” Oct. 31, 2023.
- [3] P. Cañas *et al.*, “Althena D1.1 - Methodology for Assessing the Ethics, Transparency, Accountability, and Privacy of AI-based Systems in CCAM APPLICATIONS.” Aug. 29, 2024.
- [4] M. Mitchell *et al.*, “Model cards for model reporting,” in *Proceedings of the conference on fairness, accountability, and transparency*, in FAT\* ’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 220–229. doi: 10.1145/3287560.3287596.
- [5] V. Petsiuk *et al.*, “Black-box explanation of object detectors via saliency maps,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11443–11452.
- [6] Y. Liu, T. Wang, X. Zhang, and J. Sun, “PETR: Position embedding transformation for multi-view 3D object detection,” *ArXiv Prepr. ArXiv220305625*, 2022, [Online]. Available: <https://arxiv.org/abs/2203.05625>
- [7] T. Beemelmans, W. Zahr, and L. Eckstein, “Explainable multi-camera 3D object detection with transformer-based saliency maps,” *ML4AD Workshop NeurIPS New Orleans*, 2023.
- [8] A. Radford *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.
- [9] T. Beemelmans, Q. Zhang, and L. Eckstein, “MultiCorrupt: A multi-modal robustness dataset and benchmark of LiDAR-Camera fusion for 3D object detection.” 2024.
- [10] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “The german traffic sign recognition benchmark: a multi-class classification competition,” in *Proceedings of the 2011 international joint conference on neural networks (IJCNN)*, IEEE, 2011, pp. 1453–1460. doi: 10.1109/IJCNN.2011.6033395.
- [11] G. Neuhold, T. Ollmann, S. Rotabui, and P. Kotschieder, “The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes”.
- [12] M. Alibeigi *et al.*, “Zenseact open dataset: a large-scale and diverse multimodal dataset for autonomous driving,” *ArXiv Prepr. ArXiv230502008*, 2023, [Online]. Available: <https://arxiv.org/abs/2305.02008>
- [13] R. J. Wang, X. Li, and C. X. Ling, “Pelee: a real-time object detection system on mobile devices,” *ArXiv Prepr. ArXiv180406882*, 2019, [Online]. Available: <https://arxiv.org/abs/1804.06882>
- [14] Y. Lee, J. Hwang, S. Lee, Y. Bae, and J. Park, “An energy and GPU-computation efficient backbone network for real-time object detection,” *ArXiv Prepr. ArXiv190409730*, 2019, [Online]. Available: <https://arxiv.org/abs/1904.09730>
- [15] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “Pointpillars: Fast encoders for object detection from point clouds,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 12689–12697, 2019, doi: 10.1109/CVPR.2019.01298.
- [16] T. Yin, X. Zhou, and P. Krähnenbühl, “Center-based 3D object detection and tracking,” *ArXiv Prepr. ArXiv200611275*, 2021, [Online]. Available: <https://arxiv.org/abs/2006.11275>
- [17] G. Shi, R. Li, and C. Ma, “PillarNet: Real-time and high-performance pillar-based 3D object detection,” *ArXiv Prepr. ArXiv220507403*, 2022, [Online]. Available: <https://arxiv.org/abs/2205.07403>
- [18] S. Heidrich, T. Beemelmans, A. Nekrasov, B. Leibe, and L. Eckstein, “OCCUQ: Exploring efficient uncertainty quantification for 3D occupancy prediction,” in *International conference on robotics and automation (ICRA)*, 2025.
- [19] M. Perez and A. Agudo, “Robust multimodal and multi-object tracking for autonomous driving applications,” in *2023 21st international conference on advanced robotics (ICAR)*, 2023, pp. 100–106. doi: 10.1109/ICAR58858.2023.10406433.

- [20] S. C. Davis and R. G. Boundy, "Transportation Energy Data Book: Edition 38," Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), ORNL/TM-2019/1333, Jan. 2020. doi: 10.2172/1606919.
- [21] E. Dagan, O. Mano, G. P. Stein, and A. Shashua, "Forward collision warning with a single camera," in *Proceedings of the IEEE intelligent vehicles symposium*, IEEE, Jun. 2004, pp. 37–42. [Online]. Available: <https://doi.org/10.1109/IVS.2004.1336352>
- [22] M. M. Karim, Y. Li, R. Qin, and Z. Yin, "A dynamic spatial-temporal attention network for early anticipation of traffic accidents," *ArXiv Prepr. ArXiv210610197*, Dec. 2021, [Online]. Available: <https://arxiv.org/abs/2106.10197>
- [23] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Computer vision – ACCV 2016*, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds., in Lecture notes in computer science, vol. 10114. Cham: Springer International Publishing, 2017, pp. 136–153. doi: 10.1007/978-3-319-54184-6\_9.
- [24] W. Bao, Q. Yu, and Y. Kong, "Uncertainty-based traffic accident anticipation with spatio-temporal relational learning," in *Proceedings of the 28th ACM international conference on multimedia*, ACM, Oct. 2020, pp. 2682–2690. doi: 10.1145/3394171.3413757.
- [25] W. Bao, Q. Yu, and Y. Kong, "DRIVE: Deep reinforced accident anticipation with visual explanation," *ArXiv Prepr. ArXiv210710189*, Sep. 2021.
- [26] T. J. Schoonbeek, F. J. Piva, H. R. Abdolhay, and G. Dubbelman, "Learning to Predict Collision Risk from Simulated Video Data," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, Aachen, Germany: IEEE, Jun. 2022, pp. 943–951. doi: 10.1109/IV51971.2022.9827228.
- [27] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Computer vision – ACCV 2016*, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds., in Lecture notes in computer science, vol. 10114. Cham: Springer, 2017, pp. 136–153. doi: 10.1007/978-3-319-54184-6\_9.
- [28] J. Fang, D. Yan, J. Qiao, J. Xue, and H. Yu, "DADA: Driver attention prediction in driving accident scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, pp. 4959–4971, 2022, doi: 10.1109/TITS.2021.3053484.
- [29] Y. Yao, M. Xu, C. Choi, D. Crandall, E. Atkins, and B. Dariush, "Egocentric vision-based future vehicle localization for intelligent driving assistance systems," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, IEEE, 2019, pp. 9711–9717.
- [30] Y. Yao *et al.*, "DOTA: Unsupervised detection of traffic anomaly in driving videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, pp. 444–459, 2022, doi: 10.1109/TPAMI.2021.3054887.
- [31] D. C. Moura, S. Zhu, and O. Zvitia, "Nexar Dashcam Collision Prediction Dataset and Challenge," Mar. 07, 2025, *arXiv: arXiv:2503.03848*. doi: 10.48550/arXiv.2503.03848.
- [32] W. Liu, T. Zhang, Y. Lu, J. Chen, and L. Wei, "THAT-Net: Two-layer hidden state aggregation based two-stream network for traffic accident prediction," *Inf. Sci.*, vol. 634, pp. 744–760, Jul. 2023, doi: 10.1016/j.ins.2023.03.075.
- [33] Z. Teed and J. Deng, "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow," Aug. 25, 2020, *arXiv: arXiv:2003.12039*. doi: 10.48550/arXiv.2003.12039.
- [34] M. M. Karim, Y. Li, and R. Qin, "Toward explainable artificial intelligence for early anticipation of traffic accidents," *Transp. Res. Rec.*, vol. 2676, no. 6, pp. 743–755, 2022, doi: 10.1177/03611981221091785.
- [35] H. A. Tahir, W. Alayed, W. U. Hassan, and A. Haider, "A novel hybrid XAI solution for autonomous vehicles: Real-time interpretability through LIME–SHAP integration," *Sensors*, vol. 24, no. 21, p. 6776, 2024, doi: 10.3390/s24216776.
- [36] Y. Shi, Z. Huang, Y. Yan, N. Wang, and X. Guo, "TSTTC: a large-scale dataset for time-to-contact estimation in driving scenarios," *ArXiv Prepr. ArXiv230901539*, 2023.
- [37] H. Winner, S. Hakuli, F. Lotz, and C. Singer, *Handbook of driver assistance systems*. Cham, Switzerland: Springer International Publishing, 2014.

- [38] T. Sun *et al.*, “SHIFT: a synthetic driving dataset for continuous multi-task domain adaptation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2022, pp. 21339–21350. doi: 10.1109/CVPR52688.2022.02068.
- [39] F. Yu *et al.*, “BDD100K: a diverse driving dataset for heterogeneous multitask learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2020, pp. 2633–2642. doi: 10.1109/CVPR42600.2020.00271.
- [40] A. Iglesias, M. García, N. Aranjuelo, I. Arganda-Carreras, and M. Nieto, “Analysis of point cloud domain gap effects for 3D object detection evaluation,” in *Proceedings of the 19th international joint conference on computer vision, imaging and computer graphics theory and applications - volume 4: VISAPP*, SciTePress / INSTICC, 2024, pp. 278–285.
- [41] H. Cui *et al.*, “Deep kinematic models for kinematically feasible vehicle trajectory predictions,” in *2020 IEEE international conference on robotics and automation (ICRA)*, 2020, pp. 10563–10569. doi: 10.1109/ICRA40945.2020.9197560.
- [42] F.-C. Chou *et al.*, “Predicting motion of vulnerable road users using high-definition maps and efficient ConvNets,” in *2020 IEEE intelligent vehicles symposium (IV)*, 2020, pp. 1655–1662. doi: 10.1109/IV47402.2020.9304564.
- [43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Jun. 2018.
- [44] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE international conference on computer vision (ICCV)*, Oct. 2017.
- [45] O.-R. A. D. (ORAD) SAE Committee, *Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles*, J3016\_02104, Apr. 2021. doi: [https://doi.org/10.4271/J3016\\_202104](https://doi.org/10.4271/J3016_202104).
- [46] M. Werling, *Ein neues Konzept für die Trajektoriengenerierung und-stabilisierung in zeitkritischen Verkehrsszenarien*, vol. 34. KIT Scientific Publishing, 2014.
- [47] K. Renz, K. Chitta, O.-B. Mercea, A. S. Koepke, Z. Akata, and A. Geiger, “PlanT: Explainable planning transformers via object-level representations,” in *Conference on robotic learning (CoRL)*, 2022.
- [48] H. Caesar *et al.*, “NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles.” arXiv, Feb. 2022. doi: 10.48550/arXiv.2106.11810.
- [49] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Conference on robot learning*, PMLR, 2017, pp. 1–16.
- [50] E. Gelder, M. Buermann, and O. Camp, “Coverage metrics for a scenario database for the scenario-based assessment of automated driving systems,” Oct. 2024, pp. 1–8. doi: 10.1109/IAVVC63304.2024.10786405.
- [51] T. Liu, C. Wang, Z. Yin, Z. Mi, X. Xiong, and B. Guo, “Complexity Quantification of Driving Scenarios with Dynamic Evolution Characteristics,” *Entropy Int. Interdiscip. J. Entropy Inf. Stud.*, vol. 26, no. 12, p. 1033, Nov. 2024, doi: 10.3390/e26121033.
- [52] R. Girgis *et al.*, “Latent Variable Sequential Set Transformers For Joint Multi-Agent Motion Prediction.” arXiv, Feb. 2022. doi: 10.48550/arXiv.2104.00563.
- [53] L. Feng, M. Bahari, K. M. B. Amor, É. Zablocki, M. Cord, and A. Alahi, “UniTraj: a unified framework for&nbsp;scalable vehicle trajectory prediction,” in *Computer vision – ECCV 2024: 18th european conference, milan, italy, september 29–october 4, 2024, proceedings, part XII*, Milan, Italy and Berlin, Heidelberg: Springer-Verlag, 2024, pp. 106–123. doi: 10.1007/978-3-031-73254-6\_7.
- [54] P. Sukennik and L. Kautzsch, “CoExist - D2.3 - Default behavioural parameter sets for Automated Vehicles (AVs).” Feb. 07, 2018.
- [55] *HCM 2010: highway capacity manual.*, Fifth edition. Washington, D.C: Transportation Research Board, 2010.
- [56] B. Hillbrand *et al.*, “Althena D5.2 - Report on initial use case validation.” Apr. 30, 2024.