DISSERTATIONES
INFORMATICAE
UNIVERSITATIS
TARTUENSIS
77

MAHIR GULZAR

Addressing Real-world Scenarios via Motion Prediction in Autonomous Driving

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

77

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS
77

MAHIR GULZAR

Addressing Real-world Scenarios via Motion Prediction in Autonomous Driving

UNIVERSITY OF TARTU
Press
1632

Institute of Computer Science, Faculty of Science and Technology, University of Tartu, Estonia.

Dissertation has been accepted for the commencement of the degree of Doctor of Philosophy (PhD) in Computer Science on January 06, 2026 by the Council of the Institute of Computer Science, University of Tartu.

Supervisors

Assoc. Prof.	Naveed Muhammad Institute of Computer Science University of Tartu Estonia
Principal Lecturer	Yar Muhammad Department of Computer Science University of Hertfordshire United Kingdom

Opponents

Assoc. Prof.	Sadi Alawadi Blekinge Institute of Technology Sweden
Assoc. Prof.	Eren Erdal Aksoy Lund University Sweden

The public defense will take place on February 25, 2026 at 11:15 in Narva Rd. 18-1006.

The publication of this dissertation was financed by the Institute of Computer Science, University of Tartu.

ISSN 2613-5906 (print) ISSN 2806-2345 (pdf)
ISBN 978-9908-57-120-1 (print) ISBN 978-9908-57-121-8 (pdf)

University of Tartu Press
http://www.tyk.ee/

To my late father, Gulzar Hussain Abid
my beloved mother, Asifa Sultana,
my dearest wife, Fareeha Hafeez,
and my beautiful children Affan and Mirha.

ABSTRACT

The autonomous driving industry has evolved rapidly in the past decade. Many businesses are adopting fully or semi-autonomous solutions to solve real-life problems, e.g., delivery robots, autonomous taxis, etc. Among the two popular autonomy pipelines, i.e., modular and end-to-end, a preferred choice from early to advanced stages of development is modular autonomy stacks since they discretize the overall autonomy into smaller modules from real-world sensing to actuation. Regardless of the choice, a crucial component of any autonomy pipeline is modeling the motion of other traffic participants for effectively navigating different driving scenarios. This thesis delves into enriching the existing autonomy solutions to handle different traffic scenarios using various motion prediction methods.

In this regard, we first explore the literature on motion modeling techniques and propose a taxonomy to identify prediction methods with respect to modeling approach, modeling outputs, and situational awareness. The proposed taxonomy acts as a modeling guideline on which further research is conducted. In the follow-up contributions, we propose different motion prediction solutions to enrich existing autonomy stacks for handling complex traffic scenarios. Specifically, first, we propose an open-source give-way area navigation solution that is built on top of motion prediction for other traffic participants in the scene. Secondly, to study the broader aspect of motion prediction on the performance of the autonomy pipeline, we propose a scenario-based evaluation approach to evaluate the autonomy stack on a modular and holistic level, giving us deep insights into the inter-modular dependency of motion prediction on its predecessor and successor modules. Lastly, we propose a goal-conditioned graph attention motion prediction model that gives competitive results on leaderboard benchmarks on a public dataset.

Here, we emphasize on working with open-source stacks to leverage off-the-shelf solutions for other modules of the autonomy pipeline since most of the research done under this thesis is also applied to actual vehicle and is not limited to simulations. Overall, this thesis is a blend of theoretical and applied research. It solves real-world autonomous driving problems by addressing them via literature and using the solutions on an actual autonomous vehicle.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Acronyms

AD Autonomous Driving
ADS Autonomous Driving Stack
AVs Autonomous Vehicles

BEV Birds-Eye View

CA Constant Acceleration
CNN Convolutional Neural Network
CV Constant Velocity

DBN Dynamic Bayesian Network

GC-GAT Graph Conditioned Goal Attention
GRUs Gated Recurrent Units

LiDAR Light Detection and Ranging

MHT Multi-Headed Attention
MLP Multi-Layer Perceptron

NHTSA National Highway Traffic Safety Administration

ODD Operational Design Domain

RNN Recurrent Neural Network

SAE Society of Automotive Engineers
SOTA State-of-the-art

TOF Time-of-flight
TTC Time-to-Collision

VOI Vehicle of Interest
VRUs Vulnerable Road Users
VUC Vehicle Under Consideration

LIST OF ORIGINAL PUBLICATIONS

1. **M. Gulzar**, Y. Muhammad and N. Muhammad, "A Survey on Motion Prediction of Pedestrians and Vehicles for Autonomous Driving," in IEEE Access, vol. 9, pp. 137957-137969, 2021.
   Doi: 10.1109/ACCESS.2021.3118224
2. **M. Gulzar**, Y. Muhammad and N. Muhammad, "Navigating Roundabouts and Unprotected Turns in Autonomous Driving," in IEEE Transactions on Field Robotics, vol. 1, pp. 27-46, 2024. Doi: 10.1109/TFR.2024.3421389
3. **M. Gulzar**, T. Matiisen and N. Muhammad, "Scenario Driven Development for Open Source Autonomous Driving Stack" in IEEE International Conference on Emerging Technologies and Factory Automation, 2024. Doi: 10.1109/ETFA61755.2024.10710800
4. **M. Gulzar**, Y. Muhammad and N. Muhammad, "GC-GAT: Multimodal Vehicular Trajectory Prediction using Graph Goal Conditioning and Crosscontext Attention " in IEEE Robotics and Automation Letters, vol. 10, no. 8, pp. 8316–8323, 2025. Doi: 10.1109/LRA.2025.3585757

In Publication I **[P1]**, the author proposed a novel taxonomy for categorizing motion prediction literature. Here, author reviewed different approaches of motion modelling for pedestrians and vehicles in context of autonomous driving. The author was responsible for writing all major sections of the paper.

In Publication II **[P2]**, the author proposed an open-source yielding algorithm for navigating roundabouts and unprotected turns. The proposed approach uses behavioral intersects of ego-vehicle with other vehicular agents in the scene for yielding decision-making. The author devised the complete algorithm, performed all experiments (simulation and real-world) and compared the approach with existing related work. The author wrote all major sections of the paper.

In Publication III **[P3]**, the author proposed a scenario-driven development and evaluation approach for evaluating the overall performance of the autonomy stack. Here, two feature updates including perception and motion prediction model were used as case study for evaluation cycle. Author implemented the complete evaluation cycle along with motion prediction feature update. The author was also responsible for writing all major sections of the paper.

In Publication IV **[P4]**, the author proposed a novel State-of-the-art (SOTA) multimodal trajectory prediction model conditioned on lane graph goals. Here, the author performed various modeling experiments to improve the benchmarking performance of the trajectory prediction model. The proposed architecture was built on encoder-interactor-decoder format where the interactor part was the crux of the proposed model. The author did ablation studies for validation of correctness of the model. The benchmarking was performed and results were compared with SOTA on NuScenes dataset. The author was also responsible for writing all major sections of the paper.

1. INTRODUCTION

Road safety remains a crucial concern globally, with traffic accidents accounting for a significant number of deaths and injuries every year. According to the World Health Organization (WHO), approximately 1.3 million people lose their lives annually due to road traffic accidents, and an additional 20 to 50 million people suffer non-fatal injuries.[1]. The Road Safety Annual Report 2023 further reflects these statistics, that is, despite advancements in vehicle safety, human error continues to play a major role in road traffic accidents worldwide [2]. This is also reflected by studies that indicate approximately 94% of accidents are directly attributable to mistakes made by human driving behaviors. [3]. These include behaviors such as distracted driving, speeding, driving under the influence of drugs, fatigue, and failure to obey traffic rules. With the latest advancements in technology, the list of distractions has grown. For example, mobile phones, vehicle infotainment systems, etc., have only intensified the problem. The National Highway Traffic Safety Administration (NHTSA) identifies distracted driving and weak decision-making as significant contributors to fatal crashes, emphasizing the limitations of human attention and judgment in real-time road conditions [3].

The promise of autonomous driving fundamentally stems from its ability to address the issue of human error. Beyond just road safety, it offers the potential for increased efficiency and enhanced mobility for individuals who are unable to drive due to age or disability. As these technologies continue to develop, we see their potential positive impact more and more on our daily lives, which makes the argument grow stronger that autonomous driving is a disruptive technology that could improve road safety by eliminating human errors. Arguably, with autonomous systems, the risks originating from human error can be significantly reduced, if not entirely eliminated.

To understand the autonomy itself, we refer to levels of autonomy proposed by the Society of Automotive Engineers (SAE) [4], which scale the performance of autonomy from level 0 to level 5. Here, level-0 means no autonomy, and level-5 refers to being fully autonomous with no operational design domain (ODD) constraints. Thus, full autonomy can be described as a vehicle's ability to navigate the environment from any one point to another without any human driver [5]. From a human's perspective, it is somewhat intuitive to predict the actions of other agents in the environment, e.g., the normal behavior of nearby vehicles, pedestrians, bicyclists, etc. Not only this, humans, in general, are good at identifying the intentions and negotiating with other road users. For example, a human driver negotiating with pedestrians, understanding their intent to cross or not to cross the road [7] or a courtesy give-way to a non-priority vehicle in jammed traffic etc. While the latter event may be rare, even simplistic yielding scenarios can suffer from traffic conflicts and can be influenced by social norms [8].

Autonomous Vehicles (AVs) make use of multiple sensors to perceive the environment. These sensors include RGB cameras, setero-vision cameras, Time-

Table 1: Analogy of human and AV perception [9]

Human driver	Autonomous Vehicle
Be able to see	Exteroceptive perception
Know where you are	Localization
When to accelerate, slowdown etc.	Control
Know where you want to go and how	Planning
Have an idea about the area you are in	Mapping
Know how others (on the road) behave	Behaviour modeling
Be able to interact with other	Human-vehicle interaction

of-flight (TOF) sensors such as TOF cameras, Light Detection and Ranging (LiDAR), and radars etc. In some sense, we can create a one-to-one analogy of humans perception and navigation with AV's perception and navigation of the environment, as shown in Table-1 [9]. For example, humans use colored stereo eye vision to create a sense of vision and depth. An AV can mimic this using either setero-vision RGB cameras, or using multiple RGB cameras for vision and sensors like LiDARs for depth. Similarly, humans make use of artifacts (buildings, monuments) or Global Navigation Satellite Systems (GNSS) such as the Global Positioning System (GPS) to localize themselves during navigation, in the same way an AV can extract artifacts from camera images for referencing or use GPS for localization. The challenge of autonomy is processing the raw data from these analogous sensors so that meaningful and interpretable information can be obtained to achieve overall autonomy.

In light of the inherent complexities of traffic behaviors, predicting the intent or motion of other traffic participants is arguably one of the most challenging problems in autonomous driving [10]. To be able to make rational decisions while navigating, an AV needs to be aware of its surroundings temporally. This means an AV shouldn't only be able to detect and track all the obstacles in the scene but also predict plausible future estimates of where the tracked obstacles will be in future. Predicting intents and motion profiles of obstacles helps resolve uncertainties in planned navigation. This thesis revolves around the task of motion prediction in the development of AVs and its implications in solving real-life traffic scenarios.

1.1. Objectives and Contributions of the Thesis

The overall objective of this thesis is to tackle the challenges associated with modeling the motion of traffic participants; since these challenges extend beyond the motion prediction task, such as bottlenecks in the autonomy pipeline, it is necessary to study motion prediction within the overall context of the autonomy and its applications on real traffic scenarios. In this thesis, we intend to view motion prediction from the lens of both academic and applied research. Thus, most of the work done under this thesis is also applied to an actual AV. The research questions that we address are as follows:

- **[RQ1] What are current trends in motion prediction approaches from the perspective of Autonomous Driving?**
  The key idea here is to explore the literature on motion prediction and understand what type of different modelling approaches exist and see what challenges are associated with them. This would be crucial in establishing a baseline for the motion prediction model.
- **[RQ2] How does motion prediction affect real-life traffic scenarios and the autonomy pipeline?** Since an AV is constantly navigating through different critical traffic situations, one of the less explored questions regarding motion prediction is its applicability to these scenarios.
- **[RQ3] How to construct and validate motion prediction models?** This is an open-ended research question. Since we will be working on implementing motion prediction models, the process of modeling and validation will always fit in for any type of motion model we work with.

This thesis addresses the above-listed research questions with three contributions which are comprised of four research papers. These publications will be explained in detail in the latter sections of the thesis, but a short summary of individual publications and their contribution to answering the research questions is given below along with figure-1 that provides an overview of interrelation of these contributions. It is to be noted here that throughout the thesis, we will be using some terms interchangeably; for example, motion prediction, motion forecasting, trajectory and intent prediction all fall under the umbrella of predicting the future motion estimates of traffic participants. Additionally, the use of "we" in the thesis follows formal academic convention. In this context, "we" should be understood as referring to I as the sole author throughout the thesis.

- (Contribution-I **[C1]**: Addressing RQ1) To study the trends in approaches of motion prediction, this paper explores the literature w.r.t two most common traffic participants, i.e., pedestrians and vehicles. This paper classifies models and approaches of motion prediction literature by proposing a novel taxonomy that categorizes the motion prediction model on the basis of three dimensions.
  - **Modelling approach:** essentially puts a motion model into being a classical or data-oriented model.
  - **Output type:** classifies model on the basis of multi-modality and discrete intentions.
  - **Situational awareness:** which is analogous to saying what different types of environmental factors are processed by the model before it outputs motion estimate.
- (Contribution-II **[C2]**: addressing RQ2). C2 is composed of two publications P2 and P3. P2 proposes a give-way area navigation algorithm that is

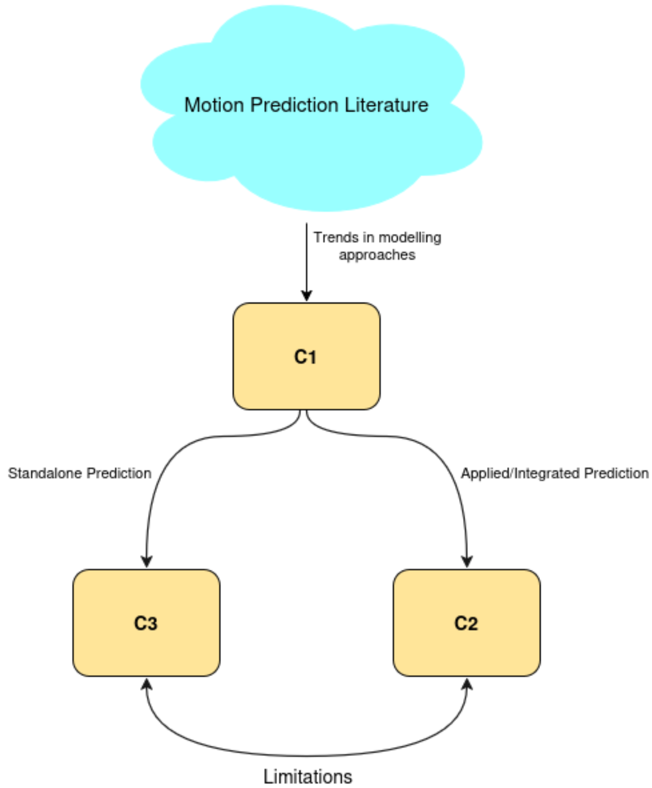


Figure 1: An overview of interrelation of thesis contributions

built on top of a multi-modal trajectory prediction model. Here, we study two real-life traffic scenarios, i.e., roundabout and unprotected turn, and apply the solution to real AV after extensively validating the model on simulated and real-world tests. Ultimately, this contribution acted as a milestone to further our progress in building navigation solutions using motion prediction. Additionally, another potential learning outcome of this contribution was the need for identification of bottlenecks from other modules of autonomy pipeline. The lessons learned from P2 were put forth into building a scenario-driven development and evaluation cycle for the autonomy pipeline in P3. In this study, we used an in-house Autonomous Driving Stack (ADS) as a case study with feature updates of a perception model and a motion prediction model highlighting inter-modular dependencies of ADS with respect to traffic scenarios. The motion prediction model in this work was evaluated using software-in-loop (SIL) and holistic evaluation, which were proposed as a continuous evaluation framework for ADS development to identify risks and shortcomings of the models before real-world tests.

- (Contribution-III **[C3]**: Addressing RQ3) In the final contribution, we propose a novel motion prediction model that uses lane graphs for vehicular trajectory prediction. The proposed model uses famous encode-aggregator-decoder [11] neural-net architecture along with goal node conditioning to forecast robust multi-modal trajectories of surrounding agents. The model is trained and benchmarked on the NuScenes [20] dataset, achieving SOTA results.

2. BACKGROUND AND FORMULATIONS

In this chapter, we will take a closer look at the evolution of motion prediction task over the years. To get a deeper understanding of its real-time applicability, we will also explore the literature on ADS and modular dependencies of traffic scenarios.

2.1. Autonomy Stacks

ADS can generally be divided in three types modular and end-to-end [5] and direct perception [6]. The modular ADS is an approach inspired by conventional autonomous-robotics approaches, where the overall problem of autonomy is discretized into smaller modules, as shown in figure-2. These have been quite popular since the beginning of research in AVs. For example, the Stanley [12] AV, winner of the DARPA challenge (often referred to as the initiation point AV challenges) in 2005 [13] used a modular ADS comprised of several modules including perception, control, vehicle interface etc. A similar modular approach was also used by Boss [14] AV, the winner of 2007 DARPA challenge [15]. Today, a popular choice in the industry is to approach AV development using modular ADS. Examples include Autoware [16] by Autoware foundation and Apollo [17] by Baidu. From an engineering perspective, solving smaller chunks of a bigger pipeline individually is more interpretable and robust, provided the inputs and outputs to and from a module are carefully engineered. Since each sub-problem in the pipeline can be assigned to specialized teams with in-depth domain-specific knowledge, it is rather easier to rigorously test, validate, identify, and troubleshoot the problems in AV's final decision-making in a modular ADS [18][19]. Multi-modularity comes with its repercussions. A bottleneck or shortcoming in one of the modules can have a cascading effect in the later modules of the pipeline. As an example, consider a scenario in which the perception module fails to detect or classify a potential object of interest (assuming it only detects specific types of objects), which later would affect the behavior of the AV. This cascading effect will follow up in the tracking, prediction, planning, and control modules of the AV, which are successor modules of perception. Another major problem of modular ADS is their lack of ability to handle rare traffic scenarios. While rare situations are a problem in general and are not specific to ADS type, since modular pipelines often employ rule-based decision-making, it is rather difficult to write rules for all possible edge cases [18].

End-to-end, on the other hand, is an emerging form of ADS that tries to solve the problem of autonomy as a whole. In contrast to modular ADS (which contains multiple sub-modules each comprised of both classical and pattern-based approaches), a neural network sits in between sensory inputs and actuation outputs in end-to-end ADS [5] as shown in figure-2. An end-to-end model relies on data i.e. given sufficient training data, the model should be able to learn an op-

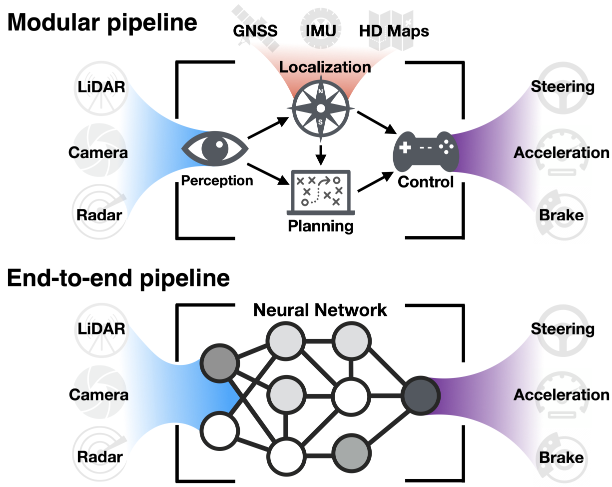


Figure 2: Modular vs end-to-end ADS [5]

timal function that drives the AV. While modular ADS can suffer from tackling edge cases and unforeseen dependencies in the scene, the end-to-end approach learns to attend to specific elements of the scene by itself. This ability to inherently attend to only relevant elements of the scene makes it a better decision-maker compared to the modular approach. On the other hand, this approach also comes with additional baggage. Firstly, these models are data hungry i.e., whether they are trained in a supervised way [20] or explorative manner [21] they require a huge amount of data for training. Secondly, end-to-end approaches are less interpretable compared to modular approaches. For example, a model could learn a wrong driving policy based on a certain bias in the training data. It is very difficult to troubleshoot why a specific wrong decision was made within end-to-end ADS.

The third often less talked about approach of autonomous driving is direct perception which is a hybrid model of ADS pioneering from [6]. It combines the strengths of the classical modular pipeline and unified end-to-end pipeline. Instead of giving network the option to output steer and control commands as it is done in an end-to-end ADS, in direction perception, the network is trained to output set of human-interpretable perception indicators that are directly relevant to the driving actions. The network outputs only a few crucial, easy-to-understand metrics called affordances. These affordances can be facts that a human driver needs to know, such as the distance to the car ahead and the angle to the center of the lane. A simple computer program then uses these facts to make the final steering and braking decisions.

In the context of motion prediction, all above approaches operate differently. Most of the modular ADS today explicitly have a motion prediction module designed precisely to predict the future motion estimates of the agents in the scene. In fact, most of the research on motion prediction models today views prediction as an individual component that can be plugged into the autonomy pipeline. For example, the open datasets on motion prediction such as NuScenes[22], Argoverse[23], Waymo[24] and lyft-l5[25], already provide you with perception, map and tracking outputs which are inputs to the motion prediction model. In comparison, the end-to-end doesn't explicitly output motion estimates; it is designed to output final actuation commands like steering, acceleration, and brake. Since interpretability is crucial in understanding the decision-making of the ADS, the end-to-end approach, in addition to actuation commands, may also output costmap, waypoints, and perception results [5]. These are often supplemented by auxiliary outputs which may or may not contain predicted trajectories of other agents in the scene. This statement is valid for direct perception too. In general, motion prediction can be seen as a hidden and implicit task of end-to-end and direct perception based ADS. Hence this research is not focused on these two types of ADS.

2.2. Scenarios in Autonomous Driving

Scenarios in AD define an evolving situation of the scene nearby the AV. They are essential in developing and validating autonomous driving systems as they involve a structured combination of conditions such as the environment, road users, and their interactions. In a sense, an AV is continuously experiencing scenarios while navigating the environment. Diverse scenarios are valuable for all modules of the ADS pipeline, such as perception, control, localization, and planning, but they are particularly important for motion prediction and decision-making since the environment is filled with dynamic, unpredictable actors. Autonomous Driving (AD) literature extensively emphasizes scenario-based validation and testing. The core concept is that scenarios provide a structured approach to identifying a range of relevant and dynamic traffic situations [30]. These scenarios are often grouped into specific categories such as lane change, emergency braking, cut-in maneuvers, and others [31]. However, this categorization is not fixed. It can be expanded and redefined based on the system under development. Scenario-based approaches allow researchers and developers to evaluate AD systems under diverse and challenging conditions, giving a deeper understanding of performance limitations and edge cases.

From the perspective of motion prediction, scenarios essentially define the operational conditions of the prediction module. This means they help identify what are the possible and impossible situations in which prediction module has to operate. Additionally, they also help in estimating the variations of the operational design domain (ODD). For example a scenario where AV gives-way to other priority vehicle in a roundabout would inherently make use of motion prediction for forecasting behavior of priority vehicle estimating how much time would it take for the priority vehicle to intersect with AV's future path. A possible variation of above ODD is an example scenario in which the priority vehicle gives way to non-priority AV due to traffic congestion. So a prediction module is stress-tested under diverse conditions by running it through an exhaustive list of scenarios with varying, environment conditions, road complexities and agent density.

2.3. Challenges of Motion Prediction

Motion prediction comes with inherent challenges which vary for different types of road users [26]. For example, a pedestrian walking in AV's vicinity may or may not use the crosswalk to cross the road. This behavior is not limited to walking pedestrians; they include cyclists/bicyclists, wheelchair users, personal mobility device users (electric scooters, hoverboards), etc. These types of road users are often referred to as Vulnerable Road Users (VRUs) [27]. The interaction of an AV with VRUs can be dictated by the traffic rules and/or social norms. For example, a human driver can infer potential future action of the VRUs based on their body gesture figure-3. The human driver can also negotiate an unmarked pedestrian crossing by giving hand gestures of his own.

Figure 3: An example of gestures used by pedestrians to negotiate an interaction [37]

These implicit/explicit negotiations with VRUs are completely non-trivial when an AV replaces a human driver. The prediction algorithm should take into account the pose, gaze, head movement, and hand gestures of other VRUs to come up with a logical decision. To be on the safe side, one could argue that the AV should be as cautious as possible, but this leads to problems like unnecessary braking, which, in addition to being inefficient, is also dangerous as the vehicles trailing the AV would be directly affected by it [26]. Additionally, there are sometimes rare events in which pedestrians can exploit this privilege. For example, several robot bullying incidents were reported in [28], where pedestrians were intentionally stopping the robots.

For vehicular agents, the challenge of motion prediction is different compared to VRUs, yet it is as complex as it is for any other type of road user. A vehicle's motion is not only affected by traffic rules; it is also affected by surrounding vehicles, i.e., an AV cannot just rely on predicting the motion of one Vehicle of In-

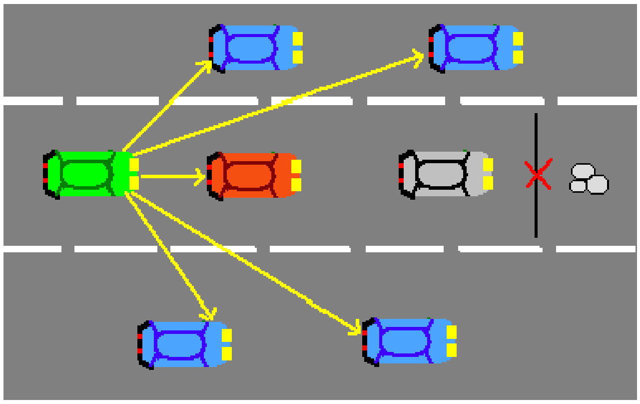


Figure 4: An illustration of how an occluded surrounding vehicle can affect the behaviour of a VOI. Here the green vehicle is the ego vehicle, the red vehicle is VOI whose future behaviour we intend to predict, the grey vehicle ahead of the red is an occluded surrounding vehicle that is about to apply emergency brakes due to a road obstruction ahead of it. [26]

terest (VOI). For example, a leading vehicle might apply emergency brakes due to another vehicle ahead of it which is occluded to AV's onboard sensors [26]. This is shown in figure-4. Such interdependency can have a cause-and-effect relation between prediction. When predicting vehicle motions, the predictions become hypotheses about how each vehicle will behave. These hypotheses interact with one another, as the decisions of one vehicle can directly or indirectly affect the decisions of others. In high-traffic environments, this interconnectedness becomes particularly important [32]. An optimal prediction result could be a joint prediction that minimizes the risk of collisions, inefficiencies, or unsafe situations for all involved. This can be modeled either as a cooperative game [33] in which vehicles work together to minimize global risk [34] or a non-cooperative competitive game in each vehicle optimizes its own trajectory with a priority on its own safety and performance [35]. In some sense, this may feel like a game theory where multiple agents (vehicles) are involved, each making optimal decisions based on the behavior of others [33] with inherent uncertainty as no vehicle can know the intentions of others perfectly, relying instead on probabilistic estimates [36].

On the bright side, since vehicular agents tend to follow traffic rules and road geometry, we can apply certain road and physical constraints to a vehicle's predicted trajectory, e.g., a vehicle will likely follow its lane and follow the curvature of the lane while turning. Similarly, a vehicle cannot abruptly turn 180 degrees, and it cannot laterally move without longitudinal motion. However, as technology advances, we may see some physical constraints become obsolete and not apply to newer vehicles [29].

2.4. Formulations

The overall task of a motion prediction model is to predict plausible future estimates for any agent in the scene. Since the model output can be of different types (which will be discussed in Chapter 3), we generalize the problem statement, assuming that we expect a motion prediction model to output multiple hypotheses or modalities. With this assumption, the problem of multimodal motion prediction for a set of dynamic agents over static contextual scene elements can be formulated as follows:

At any given instance, a scene consists of multiple traffic agents represented by their past, present, and future spatial coordinates. These coordinates for all the surrounding agents can be agent-centric and are denoted by $\mathbf{A} = \{A_0, ..., A_I\}$, where $I \in \mathbb{N}_0$ that represents total number of agents in the current scene. For each agent $A_i = (x_i, y_i, z_i, \theta_i, v_i, w_i, h_i, c_i) \in \mathbb{R}^8$, where x_i, y_i, z_i represents agent i's position in 3D space and $\theta_i, v_i, w_i, h_i, c_i$ represent yaw, velocity, width, height and class of agent respectively. Additionally, $\mathbf{A}^i_{t_h:t}$ represents a sequence of agent i's history from timestep t_h consisting of h history steps in the past up to current timestep t is input to the model. $\mathbf{M}$ represents a set of high-definition map elements that contains lanes and stop lines etc. Here, lanes are represented as $\mathbf{L} = \{L_0, ..., L_M\}$ for $M \in \mathbb{N}$ where each lane is composed of way-points and can be represented as $L_m = \{(x_{m,0}, y_{m,0}, \theta_{m,0})...(x_{m,n}, y_{m,n}, \theta_{m,n}))\}$ for $m, n \in \mathbb{N}$ and $(x_{m,n}, y_{m,n}, \theta_{m,n}) \in \mathbb{R}^3$ being the top-down 2D position and rotation of waypoint n in lane m. The stop lines are represented as $\mathbf{S} = \{S_0, ..., S_Q\}$ where $Q \in \mathbb{N}$ and $Q \not\equiv M \not\equiv I$ and $\mathbf{L}, \mathbf{S} \subseteq \mathbf{M}$. Here, $S_q = (x_q, y_q, \theta_q)$ is qth stop line's waypoint. Finally, $\mathbf{Y}^i_{t+1:t_f}$ represents a sequence of agent i's future from timestep $t+1$ to t_f. The prediction function can thus be formulated as:

$$\hat{\mathbf{Y}}^i_{t+1:t_f,k} = f\left(\mathbf{A}^i_{t_h:t}, \mathbf{M}\right) \tag{2.1}$$

where k represents number of modes or hypotheses and $k > 1$.

For a learning-based prediction model, the optimization is done via a loss function that measures the difference between the predicted values $\hat{\mathbf{Y}}^i_{t+1:t_f,k}$ and the ground truth $\mathbf{Y}^i_{t+1:t_f}$.

$$\mathscr{L} = \text{loss}\left(\hat{\mathbf{Y}}^i_{t+1:t_f,k}, \mathbf{Y}^i_{t+1:t_f}\right) \tag{2.2}$$

Since k modes represent a distribution, at least one mode should closely approximate the ground truth for successful training of a machine learning-based model.

3. TAXONOMY OF MOTION MODELS (CONTRIBUTION I)

In order to have a deeper understanding of designing and working with a motion prediction model, we need to understand different categories of motion prediction models. Various categorizations have been shown by the literature based on agent classes. For pedestrians, [38] proposes four types of categories of motion prediction models. These include models that make use of target agent's dynamics (target's motion), physiology (relative information of target w.r.t to AV), target's head orientation, and environmental context (information about map elements). A similar categorization was proposed in [39][40]. In the latter, the authors additionally added social constraints, i.e., models that consider social norms used by pedestrians while moving in groups.

For vehicles, [41] was one of the pioneering surveys that reviewed and categorized the existing literature into three broader types. These are physics-based models that apply physical constraints on the vehicle's motion, maneuver-based, ones that output the predictions in the form of maneuver trajectories, and interaction-aware models that are essentially aware of the surrounding agents. Similarly, [42] classified a very specific category of motion prediction models which are trained using deep learning. The classification proposed here is based on the model's input, prediction methodology, and model's output. Additionally, the paper also discussed evaluation metrics for motion prediction models.

In contrast to the agent's class-specific model categorization, [43] proposed a taxonomy that encompasses both pedestrians and vehicles. It classified motion models on the basis of two broader categories, i.e., modeling approach and contextual cues. Here, modeling approaches include physics-based, pattern-based, and planning-based models, whereas contextual cues include agent cues and dynamic and static cues of the surrounding environment.

3.1. Taxonomy

C1 of this thesis extends the classifications discussed above.

Specifically, We propose a novel taxonomy that classifies the literature on motion prediction over three dimensions: modelling approach, output type, and situational awareness as shown in figure-5. The key changes we propose are as follows:

- For the modeling approach, we generalize the categorization of motion prediction models for both pedestrians and vehicles as proposed by [43]. We keep the physics-based sub-category as it is from [43] and merge the pattern-based and planning-based methods from [42] and [43] into one generic category named as learning-based methods.

- For situational awareness, we build the sub-categories on top of contextual cues from [43]. Here, the sub-categories are reduced into unaware, interaction-aware, scene-aware, and map-aware models removing agent-specific categories such as articulated body pose and semantic attributes, which in the original work were specific to pedestrians agents only.
- Lastly, the output type dimension is leveraged from [42]. Here, maneuver intention, a sub-category from the original work is generalized as an intent prediction that fits in for both pedestrians and vehicular agents.

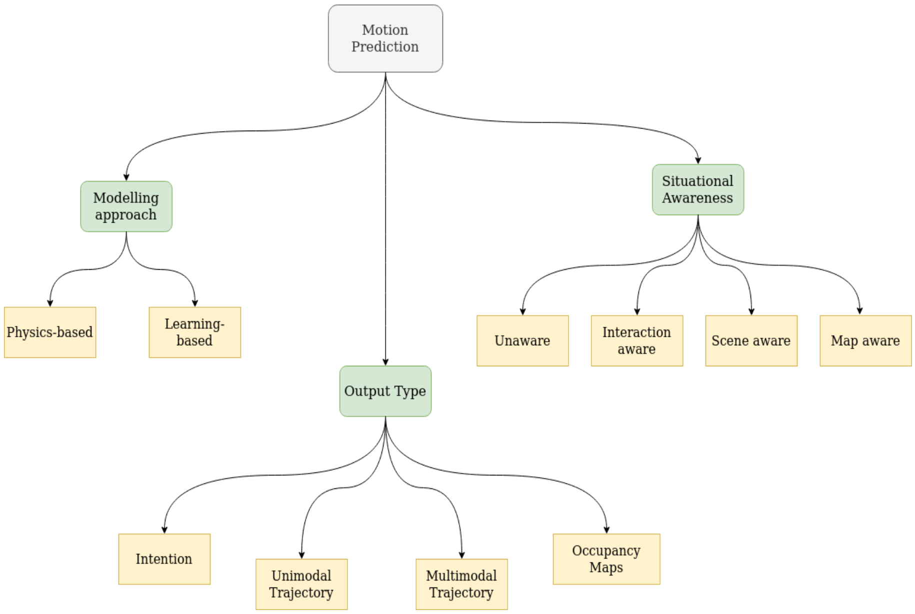


Figure 5: Proposed taxonomy of motion prediction for pedestrians and vehicles.

3.1.1. Modelling Approach

Modelling approaches are essentially the modelling techniques used to design and implement the underlying motion prediction model. In other words, this is the methodology part of any motion prediction literature. From the taxonomy we proposed in figure-5, modeling approaches are generally of two types, which are discussed in detail below.

Physics-based. As the name implies, these models are constrained on physical abilities of the moving agent, or in other words, these are governed by Newtonian laws of physics. These models can be dynamic or kinematic, where dynamic models tend to consider all forces acting upon the agent, including air friction, tire friction, forces acting inside the engine transmission for vehicular agents, etc. In contrast, the kinematic physics models are the most commonly used physics-based models in motion prediction. Since, there is no hard compulsion to include all minute complex forces acting upon the agent while predicting its future motion estimates. The kinematic models mathematically describe the motion using movement parameters. The most common family of kinematic models used today are constant X models. Here, CX can be Constant Velocity (CV) and Constant Acceleration (CA). These models take the recent relative motion of the agent into account while predicting future motion. At any given timestep t, the recent relative information of an agent can be calculated from the past two positions of the agent as shown below:

$$p = (x^t, y^t), \tag{3.1}$$

$$\Delta p = p^t - p^{t-1} \tag{3.2}$$

Here, x and y are top-down coordinates of an object (pedestrian or vehicle) and p represents the position. Hence, Δp is the most recent information to predict the future trajectory.

Apart from prediction, kinematic models are often used as process models for tracking filters. For example, [44][45] uses constant acceleration model as in Kalman filter and particle filter to predict pedestrian future motion along road semantic graph. Some works have extended CV and CA to Constant Turn Rate and Velocity (CTRV) and Constant Turn Rate and Acceleration (CTRA) to capture the agents turning non-linearity [46][47]. A further extension of kinematic physics-based models uses Interactive multiple-model trajectory prediction (IMMTP) that uses a combination of multiple constant X models for prediction estimates [48].

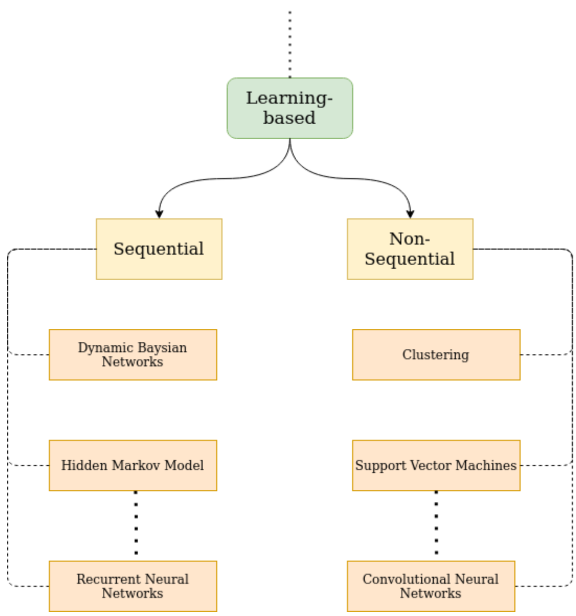


Figure 6: Categories of learning-based models. The dotted lines represent that the methods in these categories are not limited to the three methods mentioned in each category.

Learning-based. Learning-based methods are data-oriented and have an element of learning in them. These methods are comparatively better at capturing long-term dependencies and the semantics of the environment than physics-based methods. They often operate on a variety of inputs, such as collective past histories of dynamic agents, Birds-Eye View (BEV) of the environment, and raw sensor data [42]. Due to sophisticated inputs, the learning part in these methods can be a function approximator (neutral network, classical machine learning model), clustering algorithm, Dynamic Bayesian Network (DBN), or a hidden Markov model, etc. Learning-based methods are further divided into two categories as shown in figure-6.

- **Sequential learning models** have Markovian assumption which states that the future motion of an agent depends on its current and past state. These are one-step predictors and they learn from statistical observations of the data. A classic example of such models is DBNs, which use Bayesian probabilistic estimates for forecasting pedestrian and vehicular motion [49][50]. Latest works on capturing long-term dependencies have shown even more promising results. These include flavors of neural network architectures such as Recurrent Neural Network (RNN), prominently long-short-term memory (LSTM), and gated recurrent units (GRU) [51][52][53][54][55][56] [57][11].
- **Non-Sequential learning models** are not necessarily constrained over Markovian assumption, i.e., in contrast to sequential models, they don't require feedback from past frames. A classic example of such models includes clustering trajectories models that essentially group trajectories in an unsupervised manner based on the similarities of maneuver (for vehicles) or group's movement pattern (for pedestrians) [59],[60],[61],[62]. In terms of deep learning architectures, the most prominent one used for prediction tasks is the Convolutional Neural Network (CNN). CNNs are very good at capturing spatial dependencies between data points, such as image features of BEV, So CNNs are used both independently and in combinations with RNNs to capture both spatial and temporal dependencies in the input data for robust predictions [63][64][65][66][67][68].

3.1.2. Output type

A motion prediction model can be classified based on its output type. For example, a model that outputs only one modality for an agent is often referred to as an unimodal predictor, whereas having more than one output is considered a multimodal predictor. A summary of the famous output types for a motion prediction model is shown in Table-2.

The output types are divided into four broader categories. **Intention:** is high-level information of the possible current and future state of the agent. Different classes of agents can have both similar and distinct types of intentions. For exam-

Table 2: Comparison of Prediction Output types for Vehicles and Pedestrians

	Vehicles	**Pedestrians**
Intention	Turn left, turn right, go straight, changing lane etc	Will cross, not cross, walking, negotiating etc
Unimodal	Single hypothesis • Unconstrained • Constrained on maneuver	Single hypothesis • Unconstrained • Social constraints
Multimodal	Multiple hypotheses • Unconstrained • Constrained on maneuvers	Multiple hypotheses • Unconstrained • Social constraints
Occupancy maps	Occupancy predicted in future time-steps (occupancy map's resolution is crucial)	

ple, a pedestrian's intent can be "going straight," which a vehicle can also have. On the other hand, a vehicle can have an intent of "changing lane," which doesn't apply to pedestrians. Similarly, a pedestrian can have "walking" intent that doesn't apply to a vehicle. Both pedestrians and vehicles can have **Unimodal** and **Multimodal** trajectory outputs, which can be unconstrained (having no restrictions applied, e.g., traffic rules and map boundaries) and constrained (are logical with respect to nearby agents or map elements). Lastly, **Occupancy maps** are rasterized outputs of the prediction models that show possible future occupancies of the agents in the scene in the form of the top-down 2D raster. It is important to note that while the proposed output types focus on vehicles and pedestrians, it is not limited to these two agent types. It can be applied to any road user i.e. anything that moves can be represented using unimodal, multimodal, or occupancy-map–based model outputs. The only distinction required is the types of intentions. For example, a bicyclist moving through the scene cannot have a "walking" intention unless the rider is actually off the bicycle. Similarly, any other road agent not explicitly discussed in the taxonomy can still be categorized using the proposed output types by simply defining the set of applicable intentions.

3.1.3. Situational awareness

The third dimension along which we can understand a prediction model is situational awareness, which corresponds to how aware a prediction model is of its surroundings. This includes awareness about dynamic (other agents) and static (map elements) context. The situational awareness in our work is divided into four categories.

Unaware. An unaware prediction model has no scope to understand its surroundings. The only information that it processes is the history of the agent for which it is making the prediction. Unless bounded explicitly by the map constraints, most physics-based models belong to the category of unaware models.

Interaction-aware. Interaction-aware models, in addition to the history of the target agent (whose trajectory they intend to predict), use the information of other dynamic agents of their surroundings to make motion estimates. The idea behind feeding the information of surrounding agents is that the model will be able to reason the interaction of the target agent with surrounding agents before making a prediction estimate. Examples of interaction-aware models include [50][53][56][69].

Scene-aware. Scene-aware prediction models are aware of the static context of the scene. They extract contextual information of the scene using sensor data such as images. Some examples of such models include [51][52], where contextual aware pooling and top-down scene features were used to capture scene context.

Map-aware. Map-aware prediction models are somewhat similar to the scene-aware models in the sense they are also fed with information about the static surroundings of the environment. The difference lies in the format of inputs. Scene-aware models extract features and semantics from the raw sensor inputs, whereas map scene semantics can be directly given to map-aware models; thus, feature extraction purely works on pre-processed semantics given to it. Furthermore, it is not necessarily required to extract features from the map semantics; for example, for vehicular trajectory prediction, a physics-based motion prediction model can be constrained over the map lanes; all that is required here is to assign specific lanes to the target agent and roll-out the trajectories over the lanes. Some examples of map-aware models include [49][66][70][71]. One potential flaw of map-aware models over scene-aware models is that they lack information about the environment, e.g., weather conditions that are crucial in extreme weather for prediction.

3.1.4. Associated Challenges/Limitations

Apart from the categorization of motion prediction models, there are inherent underlying challenges & limitations associated to each dimension of proposed taxonomy. These listed in Table 3, 4 and 5.

Table 3: Challenges associated with modeling approach

Modeling Approach	Challenges
Physics-based	Strong constraints and assumptions about agent kinematics, difficult to generalize over complex scenarios.
Learning-based	Dependency on annotated datasets; prune to overfitting to datasets they are trained on; difficult to adapt to real-time autonomy stacks.

Table 4: Challenges associated with different output types in motion prediction

Output Type	Challenges
Intention	Gives only high level understanding of the possible future of the agent i.e. future motion profile is absent.
Unimodal trajectory	Fails to take into account the uncertainties and other highly likely scenarios which the dynamic agent can be in future.
Multimodal trajectory	Can suffer from mode-collapse (classical problem in multi-modal prediction).
Occupancy maps	High computational cost.

Table 5: Challenges associated with different situational awareness levels in motion prediction

Situational Awareness	Challenges
Unaware	No interactions are modeled between agents.
Interaction-aware	Modeling gets complex in multi-agent environment. Difficult to model in physics-based approach on the other hand in learning-based setting could induce prediction biases towards unnecessary agents in a complex scenario.
Scene-aware	Needs accurate scene understanding; perception errors can lead to ambiguous results.
Map-aware	Requires high-definition maps with lane level information depending on the agent type. i.e. sidewalks for pedestrians and road lanes for vehicles. Fail badly over unmapped areas.

3.2. Summary

In this chapter, we summarized C1, which proposes a novel taxonomy of categorizing motion prediction models based on three modeling dimensions; Modelling Approach, Output type, and Situational awareness. The categorization gave an in-depth insight into understanding motion prediction models in general. The new taxonomy proposed is AD-oriented, i.e., we used AD as the application domain while designing the taxonomy. Additionally, the proposed taxonomy is not class-independent, which means that it can be applied to almost all the dynamic object types encountered using daily AD. [RQ1] of thesis (that deals with categorizing the types of motion models) is prominently dependent on this chapter. Moreover, the categorization also helps in understanding what are the limitations to each type of motion model as discussed in previous section. This helps in establishing the bases for answering [RQ2] and [RQ3]. For example in case of [RQ2], in order to establish the affect of motion prediction on a real autonomy pipeline, we either need to employee an existing of-the-shelf physics-based solution or train a learning-based model on custom preprocessed dataset. Similarly, for [RQ3] we get a general understanding of what type of motion modeling approaches we could use as baseline and build our own model on top of it.

4. MOTION PREDICTION AND TRAFFIC SCENARIOS (CONTRIBUTION II)

In an ADS, the prediction module's output serves as a crucial input for the planning module. The planning module relies on these predictive insights to navigate complex decision-making processes, particularly when responding to dynamic and often unpredictable traffic environments. In essence, the prediction module assists the planning module by providing anticipated movements of other road users, enabling the ADS to make proactive and adaptive decisions in real-time scenarios. Accurate predictions allow the planning module to handle various traffic situations better, ensuring smoother, safer, and more reliable operations.

This contribution is directly linked to [RQ2] of the thesis i.e. to understand how motion prediction influences real-life scenarios and the overall autonomy pipeline, this chapter summarizes two publications, P2 and P3, which collectively address this research question. Specifically, P2 studies the role of motion prediction is solving a subset of real-life scenarios which AV encounters whereas P3 addresses the impact of motion prediction updates and prediction-dependent scenarios (from upstream to downstream modules of ADS) on performance of autonomy pipeline.

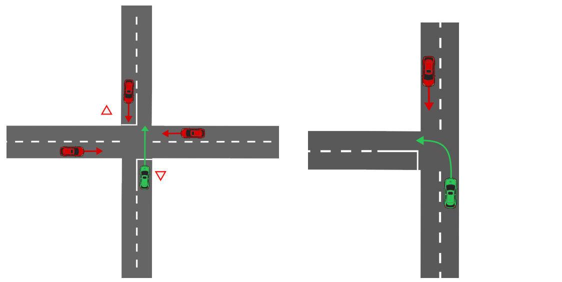
4.1. Give-way Area Navigation for Vehicles

One of the most common challenging traffic scenarios that an AV often experiences is giving way or yielding to other traffic participants. Giving way in traffic is often regulated by give-way sign, which belongs to the category of priority sign B-1 [72].

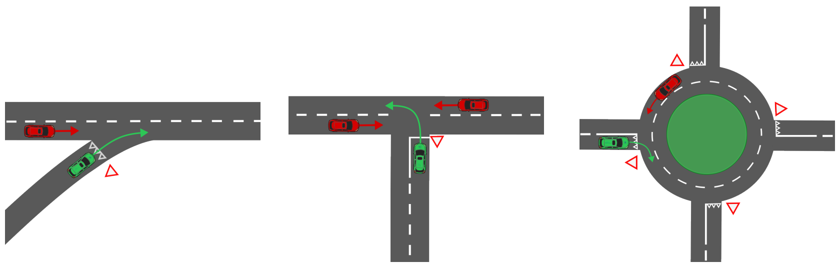
Figure-7 displays some of these typical yielding situations frequently observed in our everyday traffic. Figure-7a depicts a four-way intersection showing that the protected horizontal lane has precedence over the AV in the green. In figure-7b, an unprotected left turn is shown with no explicit yielding sign; here, the AV must yield to the oncoming vehicle. Similarly, examples of a highway lane merge and a T-junction are shown in figure-7c and figure-7d, where AV should be giving way to priority vehicles. Finally, figure-7e shows a typical roundabout situation, where vehicles entering the roundabout must yield at each entrance and merge into the roundabout without disrupting traffic.

Motion prediction is one of the primary inputs when making such navigation decisions. In such regards, first part of C2 proposes a solution for navigating yielding scenarios. The main contributions of this work are as follows:

1. Proposition of a general open-source solution to yielding problem that is implemented and integrated into Autoware.AI [16]. The solution iteratively filters VOIs from the scene using predicted trajectories of all the agents in the scene and outputs a binary yielding decision.



(a) 4-Way intersection (b) Unprotected left turn



(c) High-way merge (d) T-junction (e) Roundabout

Figure 7: Illustration of various give-way areas. Here AV is denoted by green color which does not have the right of way

2. Performance evaluation of multiple yielding scenarios in both simulation and in the real world.

4.1.1. Approach

The formulations from equation-(2.1) can be modified so that the final output is binary yielding decision instead of future trajectory points.

$$f\left(\mathbf{A}^{i}_{t_h:t}, \mathbf{M}\right) = \begin{cases} 1, & \text{if any other vehicle(s) has priority} \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

To implement the yielding decision maker based on equation-(4.1), we leverage an Open-source integrated planner of Autoware.Ai called OpenPlanner [73]. OpenPlanner gives the planning path of the ego vehicle and predicted trajectories of the agents. These trajectories are multimodal map-based predictions. The planner uses a higher-level behavior state machine transitioning from one state to another, depending on the scenario. We extend the state machine by adding two states *YieldStop* and *YieldWait* as shown in figure-8. The proposed state transition for yielding works as follows:

AV transitions from *Follow* or *Forward* state to *YieldStop* state and starts decelerating to zero velocity when there is a potential vehicle to yield. If the potential vehicle to yield hasn't crossed the yielding area and AV comes to a full stop, then

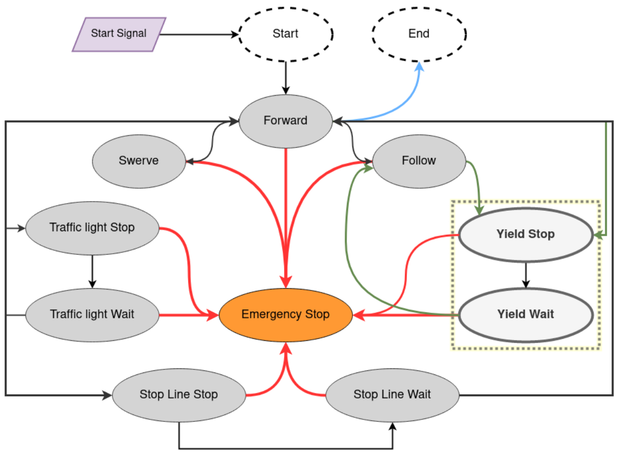


Figure 8: Behaviour state machine of AV adapted from [67] where proposed new states are shown in dotted highlighted square.

AV transitions from *YieldStop* to *YieldWait*, where AV will stay until there is no VOI to yield.

4.1.2. Implementation

With the proposed behavior state transitions, the implementation is divided into three steps:

1. **Yielding Area Identification:** As a first step, the ego vehicle should be able to identify if it is in the vicinity of the yielding area. For this, OpenPlanner gives us a rich match API from which we can query yielding stoplines in the future local navigation plan of the ego-vehicle. Once a yielding stopline is encountered within the stopline discovery distance threshold, the yielding evaluation starts.
2. **VOI Filtering:** Assuming that the ego vehicle is in the yielding evaluation area, Ego's rollouts are evaluated against all agents in the scene. The VOIs are filtered in two steps:
   - **Route Followers:** Agents that are following the same local navigation route as the ego vehicle should be removed for yielding evaluation. Let p_0 be the nearest waypoint from Vehicle Under Consideration (VUC) in ego's route and p_1 the waypoint followed by p_0. Here, p_i represents (x_i, y_i, θ_i). Let λ be the waypoint $(\hat{x}, \hat{y}, \hat{\theta})$ consisting position and orientation of VUC. The projection of λ on p_i is as follows:

$$p_i' = p_i.\mathbf{T},$$

$$where \quad \mathbf{T} = \begin{bmatrix} \cos(-\theta_0) & -\sin(-\theta_0) & -\hat{x} \\ \sin(-\theta_0) & \cos(-\theta_0) & -\hat{y} \\ 0 & 0 & 1 \end{bmatrix}$$

p'_i shows the transformed lane waypoints p'_0 and p'_1 w.r.t. VUC's frame of reference. We can calculate the $y-intercept$ or lateral distance of VUC to ego's local route.

$$m = \frac{y'_1 - y'_0}{x'_1 - x'_0},$$

$$b = y'_0 - m(x'_0) = y'_1 - m(x'_1), \tag{4.2}$$

$$\Delta\theta = \theta'_0 - \hat{\theta} \approx \theta'_1 - \hat{\theta} \tag{4.3}$$

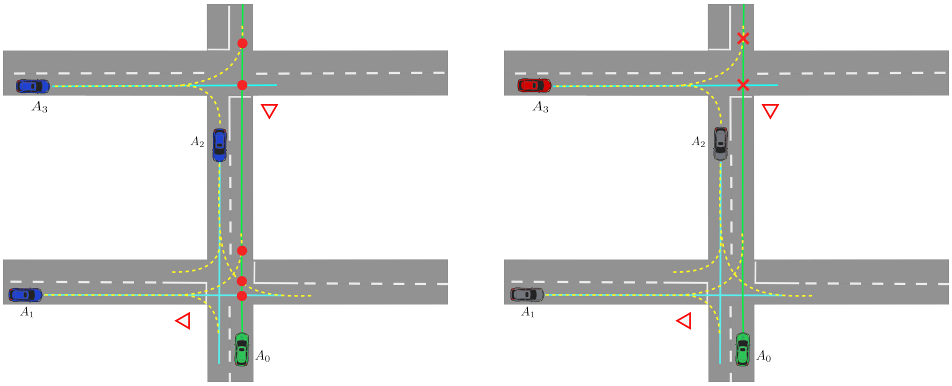
Here, b is the perpendicular distance and $\Delta\theta$ is angle difference of VUC from ego's route. An agent $\mathbf{A}^i_{t_h:t}$ is AV's route follower if $|b|$ is less than lane assignment threshold and $|\Delta\theta|$ is less than maximum yaw deviation. These thresholding parameter values are discussed later in Table-6.

- **Behavioral Intersects:** This step removes vehicle's whose trajectory intersection points are before yielding stopline to avoid unnecessary braking. OpenPlanner gives us multimodular trajectories of other agents, with discrete probabilities of each mode. For additional safety, we calculate behavioral intersections of all modes of VUC with ego since the most likely trajectory is not always the actual followed trajectory. Figure-9 shows an illustration of filtering VOIs via behavioral intersects.

3. **Decision Making:** Once yielding VOIs are isolated, we can run the yielding decision maker. The decision maker is a classical Time-to-Collision (TTC) and vehicular bounds-based metric that evaluates possible future collisions of all VOIs with ego. Let C_x and C_y be the critical lateral and longitudinal distance from the center of the ego vehicle. Moreover, let α and β be the lateral and longitudinal distance between VOI and ego at any given future time step. The collision check can then be formulated as follows:

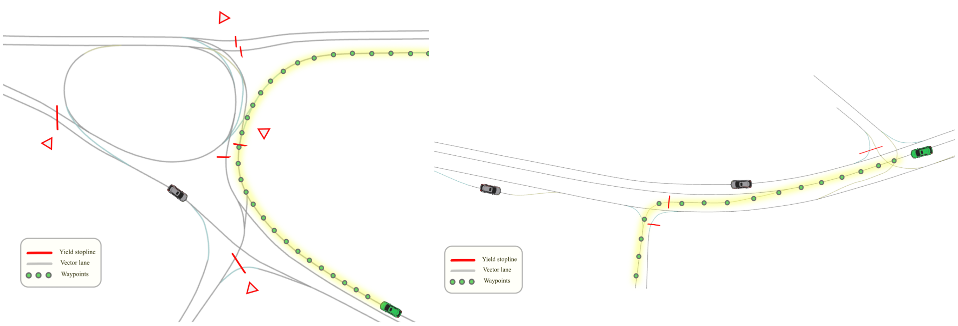
$$collision\_check(TTC, \alpha, \beta) = \begin{cases} 1, & \text{if } TTC < t_{thresh} \text{ and } |\alpha| < C_x \text{ and } |\beta| < C_y \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

Here, t_{thresh} is another safety parameter empirically tuned discussed in Table-6. *collision_check* is the final step of yielding evaluation that is responsible for the binary decision, which is referred to in equation-(4.1). The code base of this work is available at github.com/MahirGulzar/autoware_ut.



(a) (b)

Figure 9: Example of VOI filtering using behavioural intersects. Here green vehicle is ego vehicle, blue vehicles are vehicles before behavioural intersect filtering, gray vehicles are vehicles which are ignored for yielding and finally red vehicle is the vehicle which will be given right of way.



(a) (b)

Figure 10: Here (a) shows an entry point of roundabout yielding area and (b) shows an unprotected left turn in the demo route testing track.

4.1.3. Experiments

The proposed yielding area navigation is intended to be tested on an actual vehicle; consequently, the evaluations performed are carefully devised in a hierarchical manner, i.e., extensively experimenting with customized scenarios in simulation before testing anything in real traffic. We pick two yielding areas on a demo route around the University of Tartu Delta building. These yielding areas include a roundabout and an unprotected left turn, as shown in figure-10. The hierarchical evaluation of the proposed approach is divided into following:

- Simulated obstacles
- Simulated obstacles on test site
- Real obstacles on test site

Table 6: Empirically tuned, yielding decision-making parameters.

Parameter	Value	Description
lat_{safety}	0.5m	Horizontal safety box distance.
$long_{safety}$	3.5m	Vertical safety box distance.
C_x	$(\frac{ego\_width}{2}) + lat_{safety}$	Critical lateral distance from other vehicle referred from equation-(4.4).
C_y	$(\frac{ego\_wheel\_base\_length}{2}) + (\frac{ego\_length}{2}) + long_{safety}$	Critical longitudinal distance from other vehicle referred from equation-(4.4).
t_{thresh}	11s	TTC threshold.
lat_{skip}	20m	Trajectory point's lateral skip distance; beyond this, a trajectory point is not evaluated as a collision point.
max_lane_dist	2m	Maximum perpendicular distance of VUC from ego's route, after which it should be evaluated for yielding. Referred to as lane assignment threshold earlier.
$max\_yaw\_deviation$	15^o	Maximum yaw angle difference between obstacle's heading and ego's route.

- Real traffic

The evaluation process starts in a simulation setting where we have devised a set of six generic scenarios that range from easy to extreme difficulty. These scenarios are described in Table-7. Since the proposed approach gives a binary decision for yielding, the success and failure can easily be quantified. We mark a scenario to be successfully passed when ego-vehicle navigates the yielding area without colliding with any vehicle with priority. Additionally, if the ego-vehicle yields unnecessarily (false positive), the number of times unnecessary brakes were applied is added to the evaluation, too.

Simulated obstacles: Simulation experiments are listed in Table-8. We ran 12 tests per scenario for six scenarios which amounts to a total of 72 tests for each type of yielding area. The results we got from simulation shows 100% success rate with barely any unnecessary braking. The reason of such a promising result is the fact that there were no false negatives and false positives detections since the coordinates of every surrounding vehicle is known to the yielding algorithm.

Simulated obstacles on test site: In this setting, we simulated the same sce-

Table 7: Scenarios of roundabout and unprotected left turn navigation in demo route (Referred from P2)

Scenario	Description
1	Approaching yielding area, No leading/trailing vehicle. At Least 1 moving vehicle to yield.
2	Approaching yielding area, No leading/trailing vehicle. At Least 2 or more moving vehicles to yield.
3	Approaching yielding area, No trailing vehicle. At least 1 leading vehicle ahead of us before the yielding stop line and at Least 1 moving vehicle to yield.
4	Approaching yielding area, No trailing vehicle. At least 1 leading vehicle ahead of us before the yielding stop line and at Least 2 or more moving vehicles to yield.
5	Approaching yielding area, At least 1 leading vehicle ahead of us before the yielding stop line, At least 1 trailing vehicle behind us and at least 2 or more moving vehicles to yield.
6	Approaching yielding area, At least 1 leading vehicle ahead of us before the yielding stop line, At least 1 trailing vehicle behind us and at least 2 or more static vehicles to yield in the forming a traffic congestion.

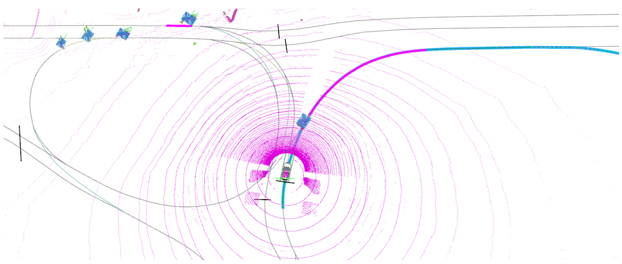
narios as discussed in the previous section with the actual ego vehicle on the test site. This means that an actual vehicle was used to navigate around a testing area with virtual obstacles. The results are listed in Table-9. Here, we ran 2 tests per scenario for six scenarios which amounts to a total of 12 tests for each type of yielding area. Here, the results are similar to previous evaluations i.e. no failures mainly due to predefined virtual obstacles with known locations. The yielding algorithm was able to take into account every vehicle in each frame without any false negatives.

Real obstacles on test site: In this setting, two real vehicles participated in the evaluation i.e. an ego-vehicle and a target obstacle vehicle. Since, there is only one obstacle vehicle, not all scenarios listed in Table-7 could be evaluated instead in almost all tests ego-vehicle was non-priority vehicle which had to give-way to the other vehicle. Additionally, the success of a test here is determined if there is no safety driver intervention in the ego-vehicle. (Safety driver only takes control if he/she thinks that there could be potential collision). Table-10 shows the results of tests. We ran 12 tests for each yielding area type. In comparison to previous evaluations, we see a significant drop in number of successes for unprotected left turn scenarios. These failures are a result of mis-detections of the object detector. Since there was no object detected in the front of the ego-vehicle, the ego-vehicle didn't explicitly yield for the oncoming vehicle and for many attempts the safety driver took over the ego-vehicle.

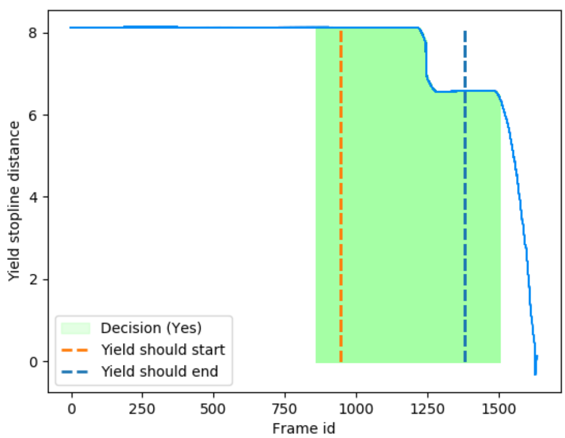
Real traffic: In final settings, we ran multiple tests on recorded data over real-traffic. The results are summarized in Table-9 which are very similar to results in Table-8. In essence, the perception module suffers from mis-detections and false positives in real data, which has a cascading effect on the overall performance of the yielding decision-making. The success and failure of these tests are determined by positive and negative yield decisions within a dedicated time frame window which is manually marked as ground truth for evaluations.

Figures-11,12 and 13 show qualitative results of the evaluations performed with real-obstacles. Here, a) shows the scene's snapshot of front-facing camera on the ego-vehicle b) shows a top-down view of the scene in with HD-map and detected obstacles whereas c) shows distance to yielding stopline per frame, the two dotted vertical lines represent manually estimated yielding ground-truth, i.e., the yielding decision should be positive within these bounds. It is to be noted that we see some unnecessary braking in figure-12 outside yielding bounds which appear from perception's false positive detections. On the other hand, in figure-13 the yielding decision within the bounds is mostly negative marking the scenario to be unsuccessful since we didn't yield inside the estimated yielding bounds. This is also caused by perception but in this case its false-negative detection that led to false-negative decision in yielding.

(a)



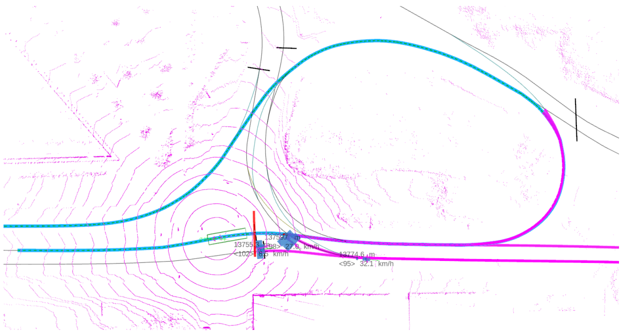
(b)



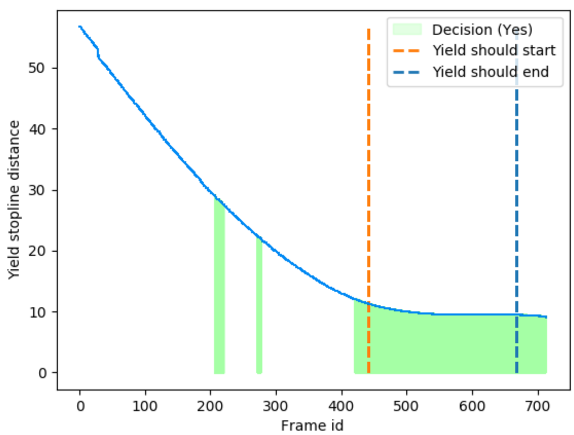
(c)

Figure 11: An illustration of qualitative results of the yielding algorithm with real obstacles at roundabout entry on test-site. Here, a) shows scenes snapshot b) shows a top-down view with perception and HD-maps whereas c) shows distance to yielding stopline per frame, the two dotted vertical lines represent estimated yielding ground-truth, i.e., the yielding decision should be positive within these bounds.

(a)

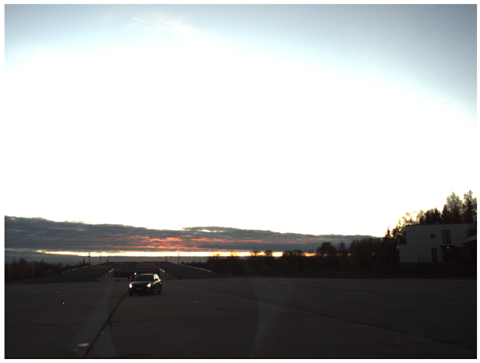


(b)

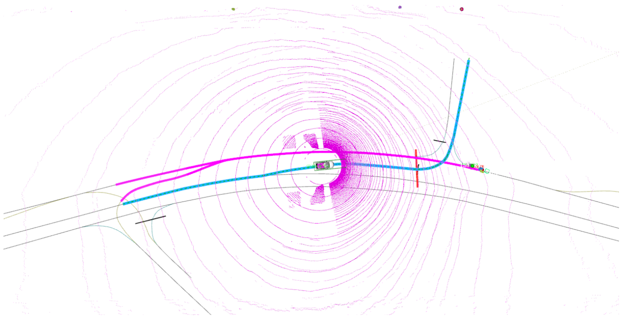


(c)

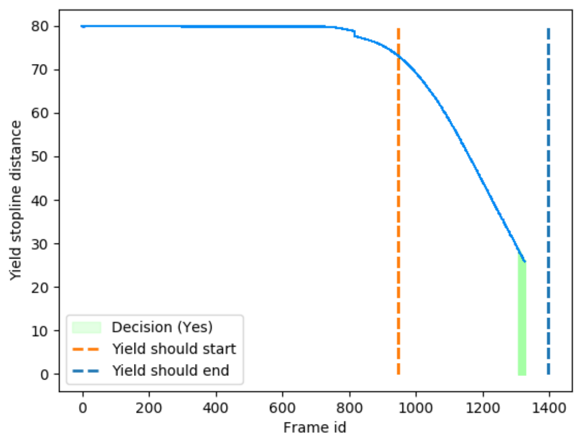
Figure 12: An illustration of qualitative results of the yielding algorithm with real obstacles at roundabout entry in real traffic. Here, a) shows scenes snapshot b) shows a top-down view with perception and HD-maps whereas c) shows distance to yielding stopline per frame, the two dotted vertical lines represent estimated yielding ground-truth, i.e., the yielding decision should be positive within these bounds.



(a)



(b)



(c)

Figure 13: An illustration of qualitative results of the yielding algorithm with real obstacles at an unprotected-left turn on test site. Here, a) shows scenes snapshot b) shows a top-down view with perception and HD-maps whereas c) shows distance to yielding stopline per frame, the two dotted vertical lines represent estimated yielding ground-truth, i.e., the yielding decision should be positive within these bounds.

Table 8: Evaluations of simulated obstacles on roundabout and unprotected left turn yield in simulation

Yielding Area	No. of Tests	Successes	Unnecessary Braking
Roundabout	72	72	1
Unprotected Turn	72	72	0

Table 9: Evaluations of simulated obstacles on test-site with actual ego-vehicle

Yielding Area	No. of Tests	Successes	Unnecessary Braking
Roundabout	12	12	1
Unprotected Turn	12	12	0

Table 10: Evaluations of real obstacles for roundabout and unprotected left turn yield on test-site

Yielding Area	No. of Tests	Successes	Unnecessary Braking
Roundabout	12	11	0
Unprotected Turn	12	4	0

Table 11: Evaluations of real traffic for roundabout and unprotected left turn yield in demo route

Yielding Area	No. of Tests	Successes	Unnecessary Braking
Roundabout	12	10	4
Unprotected Turn	12	3	0

4.2. Improving Autonomy Performance

Motion prediction can help identify risky traffic situations. For example, prediction helps avoid collisions by giving future possible trajectories to the planner, which the planner can use to plan collision-free local paths. This is very similar to what we discussed in first part of C2, where prediction helped to give way to VOIs by calculating TTC with predicted trajectories. Another similar use case of predicted trajectories is to apply emergency braking for pedestrians who jaywalk suddenly in front of the ego-vehicle. Without future motion estimates, a planner is essentially unaware of any obstacle until the obstacle comes into contact with a local path of the ego-vehicle. There are a lot of similar real-life scenarios in which motion prediction plays a crucial role.

The development of an autonomous vehicle requires these scenarios to be addressed efficiently. Moreover, every scenario tackled by the autonomy pipeline has some level of dependency on modules of the autonomy stack (Further explained in next section). These scenarios are often tackled on the development level with rigorous testing. In such regards, it is difficult to track what overall improvements have been made to the autonomy. The second part of C2 helps us answer this question, i.e., how solving scenarios helps to improve the performance of autonomy (which is also the second part of [RQ2]).

4.2.1. Inter-modular Dependencies in Autonomy Stack

Identifying module-level dependency of scenarios is critical since it is practically impossible to pinpoint why autonomy is under-performing in a particular traffic situation. Let us consider 10 NHTSA CARLA scenarios [74]. Table-12 lists these scenarios with their ADS's module-level criticality.

	Localizer	**Detector**	**Tracker**	**Motion Predictor**	**Planner**	**Controller**
Scenario-1	Critical	Not Used	Not Used	Not Used	Critical	Critical
Scenario-2	Moderate	Moderate	Critical	Not Used	Moderate	Moderate
Scenario-3	Moderate	Critical	Critical	Critical	Moderate	Moderate
Scenario-4	Moderate	Critical	Critical	Critical	Moderate	Moderate
Scenario-5	Moderate	Moderate	Moderate	Critical	Critical	Moderate
Scenario-6	Moderate	Critical	Moderate	Critical	Critical	Moderate
Scenario-7	Moderate	Critical	Moderate	Critical	Moderate	Moderate
Scenario-8	Moderate	Critical	Moderate	Moderate	Moderate	Moderate
Scenario-9	Moderate	Critical	Moderate	Moderate	Moderate	Moderate
Scenario-10	Moderate	Critical	Moderate	Moderate	Moderate	Moderate

Table 12: CARLA Leaderboard's NHTSA scenarios with potential criticality according to the modules of ADS. Here, **critical** means that the module plays a crucial role in the scenario output state, **moderate** means that the module with reasonable performance should be enough to pass the scenario successfully, and **Not Used** shows that the module is not playing any active role in passing/failing of the scenario. (Referred from P3)

Scenario-1, also defined as "Ego control loss due to bad road conditions," is critically dependent on the localizer, planner, and controller since the ego vehicle doesn't encounter any dynamic obstacle, i.e., there is no need to detect, track, and predict the motion of any dynamic obstacle in this situation.

Scenario-2, which is defined as "Leading vehicle decelerates abruptly," is critically dependent on the tracker since keeping track of the speed of the leading vehicle is the main task of the tracking module. While the tracker is crucial here, this doesn't necessarily mean that other modules are irrelevant; in fact, the tracker is dependent on detections by the perception module. Similarly, a lousy localization will also affect the lateral and longitudinal relative position of the ego-vehicle. Here, we emphasize that there is always some level of dependency of a scenario on a module that is actively participating in the decision-making process. In Scenario 2, we mark localizer, detector, planner, and controller with moderate dependency. Moreover, we mark the prediction module as not used since if predictions were absent, the tracker would still give relevant information to the planner to decelerate quickly.

Scenario-3 is defined as an "Unexpected obstacle in front of ego-vehicle" and is critically dependent on the detector, tracker, and motion predictor. Here a pedestrian walks in front of a moving ego-vehicle. The first critical thing here is to detect the pedestrian in time. Secondly, the speed of the pedestrian should be tracked well, and finally, if there is the possibility of estimating the collision with the pedestrian before even the pedestrian has entered the ego vehicle's future path, the ADS should use it. Hence, the performance of ADS can be improved in this scenario if the detector, tracker, and motion predictor play their role efficiently.

Scenario-4 is defined as an "Ego-vehicle turning left on route." Here, the ego-vehicle turning left might need to give way to another vehicle with priority, as discussed in first part of C2, so scenario 3 is critically dependent on the detector, tracker, and motion predictor.

Scenario-5 is described as an "Ego-vehicle highway merge." Here, the ego-vehicle only intends to merge into a highway and may need to give way to a priority vehicle in a specific lane. Here, the motion predictor and planner are crucial, assuming that a moderate level of performance is given by the object detector.

Scenario-6 is titled "Ego-vehicle maneuver in the opposite lane." The scenario description says that due to road blockage, the ego-vehicle needs to maneuver in an opposite lane where there is an on-coming vehicle. Here, the ego-vehicle needs to detect the on-coming vehicle in a timely manner before maneuvering into the opposite lane and giving way to the vehicle. We mark the detector and motion-predictor as critical modules of dependency in this scenario.

Scenario-7 happens on a signalized junction and is titled as "Unprotected left turn at intersection with oncoming traffic". The ego-vehicle follows the traffic signal, turns left while yielding for the on-coming traffic. Here, the ego-vehicle should detect the on-coming vehicle timely and also have good prediction accuracy to avoid crashing. We mark detector and predictor to be critical modules of dependency in this scenario.

Scenario-8 happens on an unsignalized junction and is titled as "Obstacle avoidance with prior action - pedestrian or bicycle." The ego-vehicle turns left with priority and gives way to pedestrian or bicyclist who is already on the crossing. Here, the ego-vehicle should detect the pedestrian or bicyclist timely thus we mark detector to be critical module of dependency in this scenario.

Scenario-9 happens on an unsignalized junction and is titled as "Obstacle avoidance with prior action - vehicle." The ego-vehicle turns left with priority and encounters a stopped vehicle. Here, the ego-vehicle should detect the vehicle timely thus we mark detector to be critical module of dependency in this scenario.

Scenario-10 is titled as "Pedestrian emerging from behind parked vehicle." The ego-vehicle encounters an pedestrian emerging from behind a parked vehicle and advancing into the lane. Here, the ego-vehicle should detect the pedestrian timely because as soon as the obstacle appears its on the ego-vehicle's lane and ego-vehicle should immediately apply brakes. We mark detector to be critical module of dependency in this scenario.

The assumption of a module's criticality is meant to explain the difficulty of tracing autonomy failures to their origin. In actuality, all modules of the autonomy pipeline play a crucial role in final decision-making. Having such assumptions w.r.t scenarios can assist ADS developers in designing and evaluating modular updates to the ADS iteratively.

4.2.2. Scenario-based Development & Evaluation Cycle

The University of Tartu, in 2022, started developing an in-house open-source ADS called autoware_mini [75]. In P3, we use autoware_mini as a case study to propose a scenario-driven development and evaluation strategy to keep track of the overall performance of the autonomy. The idea of continuous evaluation of ADS during development is built around how solving scenarios is essentially what an AV does whenever autonomy is engaged. During ADS development, multiple teams continuously develop different modules of autonomy, and each team proposes new updates simultaneously. In previous section, we discussed that the performance of an ADS should be validated from the lens of scenarios. For example, a new feature update in a specific module could perform well in unit tests but might fail to give better performance when tested on specific scenarios independently and holistically. For this, a scenario-driven development and evaluation on an abstract scale is shown in figure-14. At any moment of development, we can break down the steps of the proposed development and evaluation as follows:

1. A development team publishes a new feature update for an ADS module.
2. The feature goes through SIL testing. In this phase, the module is tested against module-oriented scenarios. For example, Scenario-3 from Table-12 can be used in SIL evaluations for detector, tracker, and motion predictor

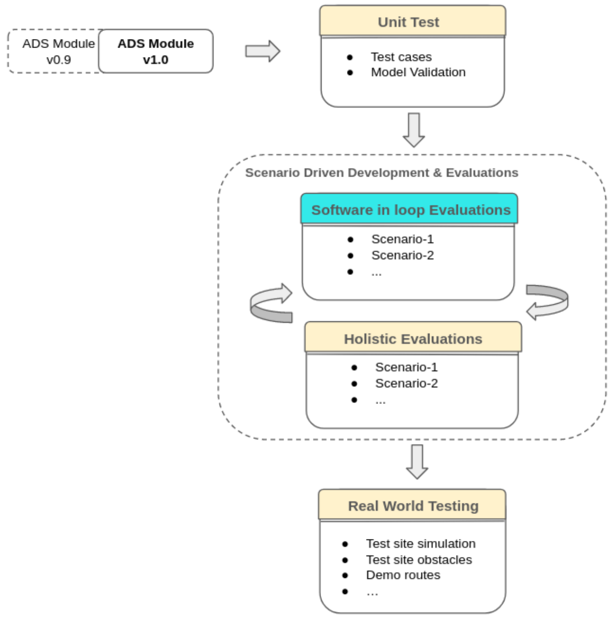


Figure 14: An abstract view of scenario-driven development and evaluation. Here, SIL evaluations are module-oriented and holistic evaluations are ADS oriented. (Referred from P3)

module updates since these modules have a critical dependency for this scenario. This evaluation keeps going until the feature is mature enough.

3. Once a development team is comfortable with the next step, the feature staged for holistic testing. In holistic testing, a set of generic scenarios is used to evaluate the overall performance of the autonomy with the new feature. In P3, we use CARLA leaderboard for getting final autonomy score. The formulations are as follows:

   Route Completion Percentage of route completed by the autonomous agent.

$$\alpha_i = \frac{C_i}{R_i} * 100,$$
$$RC = \frac{1}{N}\sum_{i}^{N}\alpha_i \tag{4.5}$$

   Here, N denotes the total number of route scenarios, R_i shows the total number of route waypoints of i^{th} route and C_i is the total number of completed waypoints by the agent on i^{th} route. α_i is percentage of completed i^{th} route and RC mean route completion score over all route scenarios.

   Infraction Penalty An aggregation of all the infractions triggered by the agent as a geometric series. Agent gets a 1.0 score with zero infractions and the score gets reduced as infractions increase.

$$\beta_i = \prod_{j}^{ped,\ldots,stop} (p_j)^{infractions_{ij}},$$

$$IP = \frac{1}{N}\sum_{i}^{N}\beta_i \tag{4.6}$$

Here, N denotes total number of route scenarios, p_j is j^{th} penalty occurred a route. β_i is the total penalty score of i^{th} route and IP is the mean infraction penalty score overall route scenarios.

Driving Score A product of route completion and infraction penalty. This is the main evaluation metric of the leaderboard i.e. one value that tells the overall performance of the ADS.

$$DS = \frac{1}{N}\sum_{i}^{N}\alpha_i\beta_i \tag{4.7}$$

Here, N denotes the total number of route scenarios and DS total average driving score over all routes.

4. The driving score with the new feature is compared with previous scores to check if the overall performance has improved or degraded. Based on this decision, a feature update will be either rejected or accepted.

4.2.3. Experiments & Evaluations

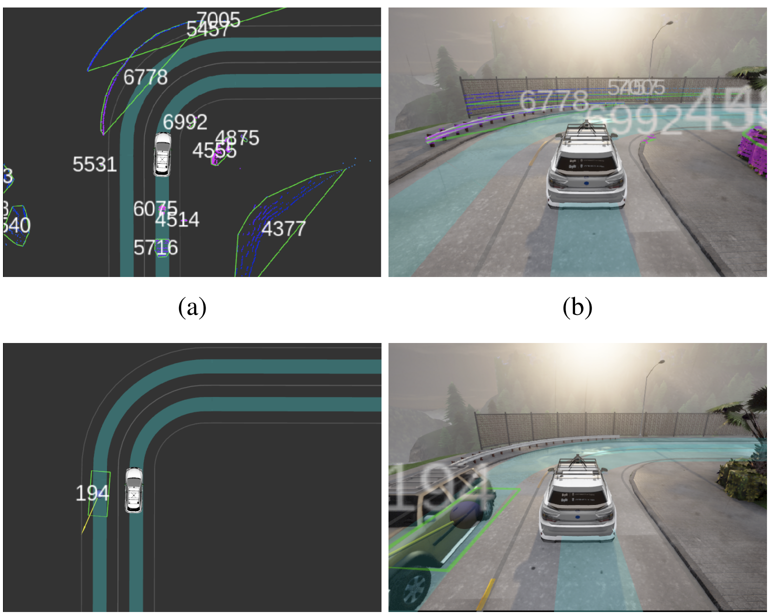
With the above-proposed development and evaluation cycle. We showed two feature updates as experiments. These updates come in the form of two models. A new 3D detector in the detection module and a naive motion predictor in the prediction module of the autoware_mini. The default 3D detector used by autoware_mini is a clustering-based detector that first removes the ground points from LiDAR point cloud and later clusters the non-ground points as obstacles. The tracker then tracks these obstacles, which are further processed by the planning module of our ADS. As a first step, the perception team proposes a new 3D detection model called SFA (Super Fast and Accurate 3D Object Detection) [76]. This is a machine learning-based model that not only detects the dynamic obstacles but also classifies them. Ideally, this model solves the problem of having so many false positives. For example, see figure-15, where both detectors are compared side-by-side. The clustering-based detector makes anything above ground an obstacle, which often marks curbs, sidewalks, rails, and buildings as obstacles. In contrast, the SFA clearly removes object cuter and only marks the passing car as a dynamic obstacle. The results are very promising when tested individually on module-oriented scenarios, and the perception team is ready to evaluate the new model against the whole pipeline.

Table 13: Clusterer vs proposed feature update's holistic evaluation score. Refer to Section-4.2.2 for scoring metrics.

Updates	DS ↑	RC ↑	IP ↑
Clusterer (Default)	**36.25**	**58.89**	**0.58**
SFA	12.73	58.88	0.18
SFA + Predictor	12.73	58.88	0.18

Alongside the above proposition, the motion prediction team proposes a naive motion prediction model for initial release. This model will predict the motion of objects classified as pedestrians. The idea here is that we intend to reduce the risk of hazards as much as possible. In this context, take the scenario shown in figure-16 as an example. When prediction is enabled, the hazard of potential pedestrian collision is identified before the pedestrian enters the ego vehicle's collision boundary. This is due to the fact that the planner took into account the potential future trajectory of pedestrians. Without prediction, we would have just identified that there is a pedestrian nearby the future local plan of ego-vehicle, but we wouldn't have identified how hazardous the situation is until the pedestrian stepped onto the road.

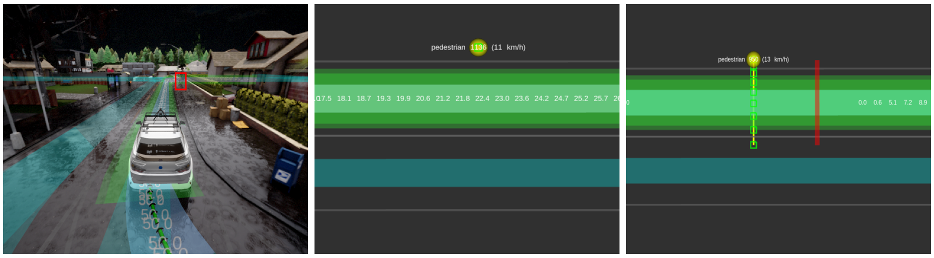
Table-13 shows the evaluation score of the ADS, where we essentially compare the default clustering detector with the proposed two feature updates above.



(a) (b)

(c) (d)

Figure 15: Cluster vs SFA model. (Referred from P3). Here, a) and b) show clustering-based detection output whereas c) and d) show SFA's detection output. The numbers are track ids of the detected objects.



(a) (b) (c)

Figure 16: An example scenario of early risk identification using using motion prediction. (Referred from P3). Here, a) shows the scenario setup scene where a pedestrian is marked with red bounding box, b) shows a top-down view of the pedestrian (yellow sphere) and ego's local path (the top lane in green) without predictor update, and c) shows a top-down view of pedestrian and ego's local path with predictor update (predicted trajectory of pedestrian intersecting ego's local path). Notice that in c) ego's path is blocked (red stop line marked as stopping point for ego-vehicle) even though pedestrian has not yet entered the collision boundary of ego's local path.

Here, we see the scenario-driven development in action, i.e., our initial module-level scenario assessments showed promising results for both of the proposed updates. Still, the holistic scores tell a different story. The overall performance of autonomy degraded with with new detector and motion predictor. By qualitatively analyzing and replaying the evaluation pipeline, it was deduced that the SFA 3D detector, though it has relatively better precision compared to clustering-based detector, but it has a very low recall for pedestrians and objects with small bounding boxes. This led to very high penalties because more collisions with pedestrians were encountered with the new model. In contrast to this, the clustering algorithm primarily detects even tiny amounts of LiDAR points above the ground as obstacles; thus, we see less collision penalization for it. In conclusion, no matter how promising the new updates look, they are postponed until a better 3D detector is introduced.

4.3. Summary

This chapter discusses the practical implications of motion prediction in solving real-life traffic scenarios and its impact on the performance of the autonomy pipeline keeping in view the [RQ2] of the thesis. In this regard, we first explored P2, where we addressed a very common yielding scenario in daily driven traffic and solved it using a multimodular prediction model that projects multiple hypotheses over a static map. The results of the yielding decision-making were affected by the perception bottlenecks of the ADS's modular pipeline. This led us to do an in-depth analysis of how the performance of the autonomy pipeline can be tracked well using AD scenarios in P3. We analyzed the inter-modular dependencies of a modular ADS w.r.t to NHTSA traffic scenarios and proposed a scenario-driven development and evaluation cycle for ADS development. We used motion prediction and perception module updates as case studies and evaluated them using the proposed evaluation framework. The chapter concludes with quantitative comparisons of autonomy performance on a holistic level.

The key takeaways for [RQ2] are as follows:

- In absence of reliable motion estimates, we couldn't have addressed the yielding decision making efficiently in P2. In general, motion prediction is crucial for solving many real-world scenarios and even naive prediction methods are sufficient which are usually available as of-the-shelf solutions. Additionally, there are some caveats with motion prediction for example, it should only be used on-demand i.e. if in P2 we evaluated predicted trajectories of other agents all the time, it would have led to unnecessary braking which is dangerous.
- Depending on the scenario which an AV is encountering, prediction module can have critical dependency on upstream and downstream modules of an ADS. Having a better prediction module doesn't always guarantee an increase in overall performance of autonomy unless upstream modules such as perception keep up with it. Same statement is valid for downstream modules too. For this reason, apart from module level (SIL) evaluation an ADS should also be evaluated holistically and scenarios are cornerstone in such type of evaluations.

5. GOAL-CONDITIONING IN MOTION PREDICTION (CONTRIBUTION III)

So far, we have addressed RQ1 and RQ2 in detail, but RQ3 requires a much deeper understanding of recent literature. To build a motion prediction model that not only does the job but can also beat the current SOTA in a public benchmark, we essentially have to see what the modeling trends are nowadays in designing motion prediction models. To get a better understanding of the latest trends, we turn towards Chapter-1 and Chapter-3. Since the boom in deep learning, a significant number of research on prediction models today is built on top of learning-based models as mentioned in Chapter-3. Early works in learning-based models include [77][78][79], which use vanilla CNN for encoding rasterized top-down semantic view of the scene. Rasterization is often compute-intensive and is not suitable for real-time applications. In 2020, a different modeling approach was proposed by VectorNet [80], in which the input is a sequence of structured vectors instead of dense rasters. This approach drastically improved computation time and the model outperformed various existing benchmarks. From VectorNet onwards, most of the works have entirely moved away from the rasterization of scenes. In very recent trends, researchers have exploited the graph structure of the road network for encoding static scene features and dynamic context [58][81][82][83][84]. Here, we see a blend of multiple neural network architectures, such as transformers, graph neural networks and recurrent networks, etc, for encoding relevant input and decoding trajectory outputs.

Keeping in view the current trends, in C3, we build a motion prediction model that is inspired by [58][82][83][84]. While these works encode both temporal and spatial scene context, they are less attentive towards target-centric goal nodes. We argue that an additional step of goal-conditioning and cross-attention will capture future dependencies better than the previous methods. In C3, We propose Graph Conditioned Goal Attention (GC-GAT): a multimodal trajectory prediction network conditioned on agent-centric graph goals.

5.1. Methodology

This section describes our proposed model with inputs and outputs to and from the model along with the main architecture of the model.

5.1.1. Problem Statement

Referring to the original problem statement of the thesis in equation-(2.1), there is a slight modification of symbols in C3, as shown below.

$$\hat{\mathbf{Y}}^{i}_{t+1:t_f,k} = f\left(\mathbf{X}{t_h}^{i}, \mathbf{C}\right) \tag{5.1}$$

$\mathbf{A}^i_{t_h:t}$ is replaced by $\mathbf{X}^i_{t_h:t}$ representing a sequence of agent history. $\mathbf{M}$ is replaced by $\mathbf{C}$, which represents static scene context vectors. The rest of the problem statement stays the same.

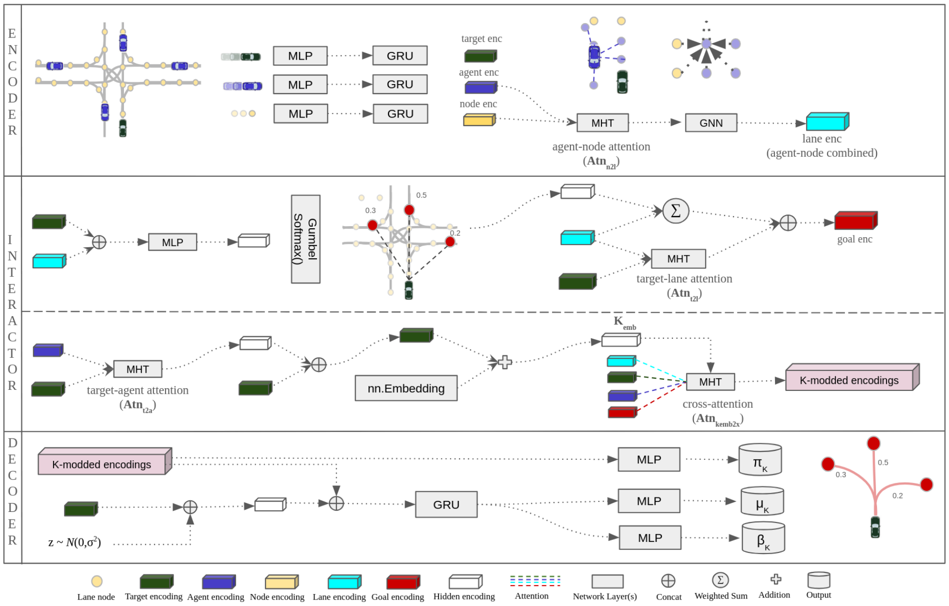


Figure 17: Model architecture of GC-GAT referred from C3. Here, the data flows from left to right in each row, and the architecture should read in the same manner, going from top to bottom.

5.1.2. Input

The proposed model's input has two parts as shown in equation-(5.1). The agent $\mathbf{X}^i$ historical data consists of $X_i = (x_i, y_i, v_i, a_i, w_i, I) \in \mathbb{R}^6$ where x_i, y_i are 2D top-down spatial coordinates of the agent v_i, a_i, w_i represent speed, acceleration, and yaw_rate. Here, I is a boolean flag for the pedestrian class. For static context $\mathbf{C}$, the HD map is represented in the form of a directed lane-node graph. Essentially, the context $\mathbf{C}$ is a super-set of $V = \{V_1, ..., V_v\}, v \in \mathbb{N}$ which is a set of lane graph nodes where each node consists of equidistant segments of lane center lines. A vth node is represented by $V_v = (x^v_n, y^v_n, \theta^v_n, I^v_n)$ where x^v_n, y^v_n, θ^v_n are spatial coordinates with yaw for the nth pose of vth's segment. Analogous to dynamic context, I^v_n is a boolean flag identifying whether lane segment's pose is inside a stopline or crosswalk.

5.1.3. Output

The output trajectory of the model is perfectly represented by equation-(2.1) and equation-(5.1) i.e. $\hat{\mathbf{Y}}^i_{t+1:t_f,k}$. To further elaborate this, the output consists of a sequence of spatial coordinates, $\hat{\mathbf{Y}}^i_{t,k} = \{(x^i_{t,k}, y^i_{t,k} | t = 1, ..., t_f\}$ where $(x^i_{t,k}, y^i_{t,k})$ is the ith trajectory point of kth hypothesis at t timestep.

5.1.4. Model

GC-GAT's architecture diagram is shown in figure-17. The model follows famous encode-Interactor-decoder architecture. These are explained as follows:

Encoder. The very first step of GC-GAT is encoding both scene features and dynamic agent histories. Both static and dynamic context has spatial coordinates along with temporal dimension. The input to the model is first processed by simple Multi-Layer Perceptron (MLP) and then handed over to lightweight Gated Recurrent Units (GRUs). Once the spatial and temporal features are encoded they are further processed by Multi-Headed Attention (MHT) layers. For example, a cross attention between other agents and nodes of the lane graph is calculated to encode the interaction of surrounding agents with the map elements. As the final step of encoding, the model exploits the graph structure of the lane nodes and applies a GNN (Graph neural network) layer on previous encoding called *lane enc* in figure-17. The encoder layers of our model can be formulated as follows:

$$
\begin{aligned}
h_{target} &= GRU(MLP(A_{target})), \\
h_{nbr} &= GRU(MLP(A_{nbr})), \\
h_{lane} &= GRU(MLP(V_{lane}))
\end{aligned}
$$

Here, h_{target}, h_{nbr}, and h_{lane} are independent hidden encoded states of the target agent, neighboring agents, and lane nodes, respectively. Once we obtain these hidden states, the encoder applies a MHT layer between h_{nbr} and h_{lane}. This captures the first cross-interaction between neighboring agents and the lane nodes.

$$
\begin{aligned}
Atn_{n2l} &= MultiHead(Q_{nbr}, K_{lane}, V_{lane}), \\
h_{lane} &= h_{lane} \oplus Atn_{n2l}, \\
h_{lane} &= GNN(h_{lane})
\end{aligned}
$$

Interactor:. The interactor is responsible for modelling interactions between different scene elements. The encoding outputs specifically *lane_enc* and *target_enc* are consumed multiple times in interactor. The hidden features of target-agent, neighbouring agents and lane are first fed to MHT layers to capture cross-attention between the dynamic and static context. These attentions are represented as:

$$
\begin{aligned}
Atn_{t2l} &= MultiHead(Q_{target}, K_{lane}, V_{lane}), \\
Atn_{t2a} &= MultiHead(Q_{target}, K_{nbr}, V_{nbr})
\end{aligned}
$$

Here, Atn_{t2l} is target-agent to lane graph attention and Atn_{t2a} is target-agent to neighbour agents attention. These attention outputs are consumed by two different branches of the model.

In the second step, a target-centric goal predictor computes goal probabilities for the lane node graph. Initially, a simple MLP is applied over the concatenated hidden encodings of the target agent and lane graph. The output encodings are then processed by the Gumbel-Softmax to obtain soft samples. h_{goal} represents the goal node encoding which are extracted by taking weighted sum of h_{lane} and gumble-soft samples. These interactor layers can be formulated as follows:

$$\mathbf{g}_{\text{enc}} = MLP(h_{\text{target}} \oplus h_{\text{lane}}),$$

$$\mathbf{g}_{\text{weighted}} = GumbelSoftmax(\mathbf{g}_{\text{enc}}, \tau),$$

$$h_{\text{goal}} = \sum_i g^{(i)}_{\text{weighted}} h^{(i)}_{\text{lane}},$$

$$h_{goal} = Atn_{t2l} \oplus h_{goal}$$

In the third step of the interactor, the target encodings are concatenated with Atn_{t2a}. Afterward, k_{emb} embeddings are added and fused with four different attention layers, each capturing k mode-oriented attention over the hidden encodings.

$$h_{target} = h_{target} \oplus Atn_{t2l} \oplus Atn_{t2a},$$

$$k_{emb} = nn.Embedding(K, D),$$

$$k_{emb} = k_{emb} + h_{target},$$

$$Atn_{k_{enc}2x} = MultiHead(Q_{k_{emb}}, K_x, V_x),$$

$$k_{enc} = norm(Atn_{k_{emb}2x})$$

Here, k_{emb} represents a trainable embedding matrix initialized randomly, with K being the number of trajectory modes and D the dimensionality of each embedding. $Atn_{k_{enc}2x}$ represent MHT layers for capturing k modded attention between k_{emb} and x, where x are hidden encodings consisting of $h_{target}, h_{lane}, h_{nbr}$ and h_{goal}.

Decoder:. The decoder is inspired from [11][82] and [84] where we make use of Laplacian mixture density network. At the first step we use an MLP to extract k-mode probabilities π_K, where π_K is merely a Laplacian mixing coefficient. Afterward, a Gaussian noise z is added to h_{target} encodings. This helps to capture longitudinal variability in the predictions. Afterwards, the aggregated encodings $h_{target} \oplus k_{enc}$ are then processed in a GRU cell for decoding the temporal dimension of the output as suggested by [82]. These network layers are formulated as follows:

$$\pi_K = MLP(k_{enc}),$$

$$h_{target} = h_{target} \oplus z \sim \mathcal{N}(0, \sigma^2),$$

$$out_{temp} = GRU(h_{target} \oplus k_{enc})$$

A Laplace distribution is represented by three parameters i.e. location μ, scale b and mixing coefficient π_K. Since π_K is already calculated in the previous step. $\{\mu_K, b_K\}$ are predicted by the two MLPs.

$$\mu_K = MLP(out_{temp}),$$

$$b_K = MLP(out_{temp})$$

Loss function:. We leverage different types of loss functions for GC-GAT. These include Laplace negative log-likelihood, cross-entropy losses for trajectory regression & mode classification and finally a classical minADE loss between predicted trajectories and the ground-truth. Thus, the equation-(2.2) changes as follows:

$$\mathscr{L} = L_{reg} + L_{cls} + L_{ADE} \tag{5.2}$$

5.2. Evaluations

Table 14: GC-GAT's Benchmarking results on *nuScenes* test set. The best scores are marked in bold and second best scores are marked with astrick an '*'. (Referred from C3)

Works	**MinADE$_5$ ↓**	**MinADE$_{10}$ ↓**	**MissRate$_5$ ↓**	**MissRate$_{10}$ ↓**	**OffRoadRate ↓**
MultiPath [85]	2.32	1.96	-	-	-
CoverNet [79]	1.96	1.48	0.67	-	-
Trajectron++ [**72**]	1.88	1.51	0.70	0.57	0.13
AgentFormer [86]	1.86	1.45	-	-	-
MHA-JAM [87]	1.81	1.24	0.59	0.46	0.07
CXX [88]	1.63	1.29	0.69	0.60	0.08
LaPred [89]	1.47	1.12	0.53	0.46	0.09
GOHOME [90]	1.42	1.15	0.57	0.47	0.04
Autobot [91]	1.37	1.03	0.62	0.44	0.02
THOMAS [92]	1.33	1.04	0.55	0.42	0.03
PGP [11]	1.27	0.94 *	0.52	0.34	0.03
XHGP [83]	1.28	0.95	0.53	0.34 *	0.03
MacFormer [93]	1.21 *	**0.89**	0.57	**0.33**	0.02
LAformer [82]	1.19	1.19	**0.48**	0.48	**0.02**
GC-GAT (Ours)	**1.19**	1.06	0.52 *	0.49	0.03 *

For C3 we turned toward public dataset to make a distinct comparison between current SOTA. Specifically, we train, test, and validate our proposed model on the NuScenes [22] prediction dataset. We employ the standard nuScenes evaluation metrics which include minimum average displacement error or point-wise $L2$ error between prediction and ground-truth over top K modes ($MinADE_K$), final displacement error over top K modes ($MinFDE_K$), miss rate over K modes ($MissRate_K, 2$) which is pointwise distance between predicted trajectory and ground-truth trajectory and finally, off-road rate, that measures the fraction of how much of the predicted trajectory is going beyond the drivable area. The benchmarking results of our model are reported in Table-14. Our model achieved a score comparable to the current SOTA on the nuScenes leaderboard.

5.3. Integration Limitations

In contrast to other contributions, this study is conducted on a clean, publicly available dataset. The next step from training and validating a model on the clean dataset is to make it usable on an actual vehicle. While GC-GAT's explicit evaluation on an autonomous driving stack hasn't been performed due to the scope of this study. The model can be adapted to a practical autonomy pipeline, but there are some caveats. Most existing works on motion prediction nowadays have the inherent problem of adaptability, i.e., adapting these models to practical autonomy pipelines suffer from performance disparity AKA *dynamics gap* [94]. Nevertheless, the existing models can be ported to any ADS, and a dynamic evaluation mechanism as proposed by [94] can be employed to evaluate real-time performance.

5.4. Summary

In this chapter, we summarized C3 of the thesis, which delves into the architectural design of the motion prediction model. In C3, we designed a motion model that treats static map lanes as road graphs and predicts trajectories over goal condition graph nodes. The model is built using different flavors of neural network layers handpicked from SOTA literature. In addition to making use of existing literature we demonstrated that goal conditioning captures future dependencies better than previous methods giving robust results. Our proposed model was rigorously evaluated on the widely recognized nuScenes dataset, and we reported competitive performance on the official public leaderboard. The results indicate that our method achieves accuracy on par with current state-of-the-art (SOTA) techniques while offering additional interpretability and robustness. Beyond the technical design and benchmarking, we also conducted a critical analysis of the practical challenges involved in transitioning such dataset-specific models to real-world autonomous driving systems. This includes discussions around deployment and the dynamics gap between offline benchmarks and online system performance. These reflections are crucial for understanding both the potential and the limitations of deploying learning-based motion prediction systems in real-world autonomy pipelines.

6. DISCUSSION

In this chapter we discuss the key contributions of the thesis. Here, we will revisit C1,C2 and C3 and highlight the strengths and limitations in an overarching manner.

6.1. Taxonomy of Motion Models

To understand different types of motion prediction models in autonomous driving, we presented a novel taxonomy that categorizes prediction models into three independent dimensions i.e. modeling approach, output type and situational awareness. Each of these dimensions gives an overall understanding of specific aspects of the prediction model. For example, modeling approach essentially identifies types of motion modeling methods. Output type lists possible model outputs independent of what modeling approach was used by the prediction model. Lastly, situational awareness lists level of awareness of a prediction model. While the proposed taxonomy covers most aspects of prediction models, one possible missing dimension is the type of input they receive. We argue that this is inherently addressed through situational awareness i.e. models rely on two forms of context dynamic and static and their awareness level depends on how richly they are informed by these inputs.

The proposed taxonomy also yielded a secondary benefit, it allowed us to assess strengths and shortcomings of prediction models w.r.t each dimension as listed in Table-3, 4 and 5. These insights were useful for informed baseline model selection for the next contributions which differ for C2 and C3, as C2 examines the integrated aspects of prediction models, whereas C3 explores them from standalone perspective.

6.2. Motion Prediction and Traffic Scenarios

With informed understanding of different types of prediction models from C1, the next step of the this thesis involved understanding how prediction help us address real world scenarios and behavior of prediction model in integrated system. In C2, we explored prediction as integrated module of the ADS. Here, we addressed one of the most common yet challenging scenarios in autonomous driving i.e. yielding for other vehicles using motion prediction by proposing a 3 step solution which was rigorously tested and evaluated in both simulation and real-world. From the taxonomy proposed in C1, we observed that although learning-based models offer stronger generalization capabilities, they were unnecessarily complex for addressing the underlying yielding scenarios. In our case, a simple kinematic model allowed us to rapidly prototype an effective yielding solution. It does not necessarily mean that kinematic models are the optimal choice overall, rather, they serve as a practical alternative until a custom-trained or fine-tuned learning-based prediction model becomes available. In our experiments, the employed prediction model in C2 can be categorized as physics-based, multimodal, unware model. Interestingly, we were able to achieve the multi-modality by simply rolling out trajectories along relevant lanes under the target vehicle assuming forward motion profile. Additionally, even though the employed prediction model isn't aware of other surrounding agents it more or less worked for yielding scenario since capturing inter-agents iteractions among other dynamic agents was not the primary objective here. One could argue that a sophisticated learning-based model would have given us same if not better yielding performance or in other words the prediction performance difference would be negligible for the specific scenarios we were trying to solve. On the contrary, the main bottleneck we faced wasn't the performance of prediction itself but the perception errors which had a trickle down effect on prediction and planning module. Moreover, we also identified absence of a scenario driven evaluation framework that could evaluate the performance of one or multiple modules of the ADS.

The above shortcomings triggered research for the second part of C2. We essentially wanted a framework in which performance of the autonomy on module level and holistic level could be evaluated using scenarios while developing our in-house ADS Autoware-mini [75]. For this, we first analyzed scenarios with respect to their criticality level on modules of autonomy. Here, we turned towards CARLA simulator and its existing leaderboard instead of developing a benchmarking mechanism from scratch. For initial tests, we worked with 10 NHTSA scenarios in the testing system. We saw proposed evaluation framework in action when two concurrent module level updates (perception and prediction model) were added in the testing system for benchmarking. The module-level tests showed significant performance gains in autonomy, whereas the holistic evaluations revealed an overall performance degradation, leading us to defer the proposed updates. This established a strong benchmarking system for ADS developers, enabling them to efficiently observe performance differences across modular updates before conducting real world tests.

From the point of view of vehicular yielding scenarios which we evaluated in P2, this system fits well, as we only need to create relevant yielding scenarios and add them to the scenario pool. The main limitation is that scenario creation is often time consuming, and we did not explicitly include the P2 scenarios in P3 due to the scope of the study. However, P3 does showcase a pedestrian emergency braking/yielding scenario as an example demonstrating usability of the framework.

6.3. Goal-Conditioning in Motion Prediction

C3 represents another informed extension of C1, similar to how C2 emerged as an informed outcome. The key difference is that in C3, prediction is treated as a standalone module rather than being restricted to an integrated system. No scenarios were involved, which gave us the liberty to work with preprocessed datasets and to evaluate our results on existing public leaderboard benchmarks. The main objective of C3 was to create and validate a prediction model. From C1 and C2, we already established that learning-based models offer stronger generalization capabilities. C1 further showed that multimodal prediction outputs are effective, provided that each mode captures a diverse hypothesis. Finally, we found that situationally aware models give a much deeper understanding of traffic compared to unaware models.

The above understanding already sketches a picture of how a baseline model for C3 would look like. In such regards, we chose [11] as our baseline which is a learning-based, map+interaction-aware, multimodal prediction model and proposed new architecture inspired from it. The proposed architecture is famous encoder-decoder architecture where an interactor module sits in between them. The major novelty of C3 comes from interactor module where we introduce goal-encodings and use them in cross-attention with other encodings essentially making the model goal conditioned. In contrast, [11] relies on graph traversals in this component. This also means that we didn't had to employee an explicit behaviour cloning loss function for training graph policy traversal header instead we directly used log-likelihood and cross-entropy losses for trajectory regression and mode classification. In addition to this, we also incorporated minADE into the loss function, thus aligning the training objective more closely with one of the key evaluation metrics of the nuScenes dataset on which the model was trained and evaluated. Moreover, to ensure that the model generates diverse outputs rather than collapsing all predictions into a single trajectory (the mode-collapse problem), we employed a winner-takes-all loss strategy. In this approach, the loss is computed only between the ground-truth trajectory and the highest-scoring predicted mode, while the other predicted modes remain independent of backpropagation. This encourages the network to spread its predictions across multiple plausible trajectories.

One additional aspect of the proposed model in C3 is that its a single-agent trajectory predictor as it is inspired from its baseline's [11] architecture (which is also a single-agent predictor). The proposed architecture would require significant modifications to make it a multi-agent predictor. The nuScenes prediction API gives agent level information in global frame-of-reference. At the moment we translate the whole top-down scene into any agent's relative frame-of-reference making the whole scene revolve around the target-agent. We would need to remove the concept of target vehicle and since networks don't generalize on absolute coordinates we would need to make an assumption of ego-vehicle (which at the moment there isn't in the nuScenes dataset) from one of the target vehicles and evaluate surrounding agents as whole at the last step. Multi-agent setting is not a part of this work.

Overall, the proposed model in C3 achieved performance comparable to the SOTA on nuScenes dataset. The proposed model outperforms the chosen baseline in minADE for top-5 highly likely trajectories. Qualitatively, our model showed diverse logical trajectories due to the element of goal conditioning. However, the model had two most prominent generalization limitations i.e. firstly, predicted trajectories sometime go over non-drivable islands and secondly, model can sometime predict illogical goals e.g. goals over wrong lanes. On the brighter side, these limitations mostly occur over less likely predicted modes. Finally, we concluded C3 with a discussion about the dynamics-gap problem, which highlights the performance disparity that can occur when models trained on preprocessed datasets are deployed on real-time integrated systems, similar to what we had in C2.

6.4. Summary of Insights

Among the three contributions of the thesis, the overarching goal was to investigate motion prediction for autonomous driving both as an integrated module and a standalone learning component which has been largely achieved. One key insight emerges from this work is that the effectiveness of a prediction model is very much dependent on its Operational Design Domain (ODD) and context. In C1, we established strengths and limitations of different types of prediction models but C2 gave us the realization that it is not about the best type of prediction model in general rather it is always about the type of prediction model that is most effective in the specific context i.e. often less sophisticated models work in real ADS pipelines. In real ADS, the errors in upstream modules specifically perception module often dominate prediction's performance. Moreover, improving one module in isolation doesn't necessarily guarantee holistic improvement in the performance of the ADS. From the perspective of standalone prediction in C3, we established that multi-modality, situational awareness, and architectural design are key contributors to the performance of standalone prediction models which is precisely what C1's taxonomy informed us. At the same time we also established that this performance factor is only effective when prediction is treated as an isolated module of the autonomy which mostly works in case of preprocessed public datasets but is not the case in real pipelines as informed by C2.

7. CONCLUSIONS

This work revolved around solving real-life autonomous driving scenarios using motion prediction module of the autonomy stack. The research conducted in this thesis enriches existing autonomy solutions and is potentially applicable to newer autonomy pipelines in general.

To address [RQ1], we conducted a profound literature review of motion prediction and introduced a novel taxonomy that categorizes models based on their input/output formats and modeling characteristics. In C1, the proposed taxonomy offered a clearer understanding of how prediction models should be systematically grouped and what capabilities can be expected from each category from the perspective of autonomous driving.

We addressed [RQ2] by applying motion prediction models to real-life and simulated scenarios, giving us a rich picture of how well motion prediction works in action and what are its potential performance blockers. In C2, we employ motion prediction to solve multiple real-life autonomous yielding scenarios, which quite frequently occur in daily driving. We proposed an open-source give-way area navigation solution on an existing Open-source ADS (Autoware), which is extensively used by the industry and academia, making our research more valuable to both applied engineers and researchers.

As a follow-up, In C2, we extended our previous work by proposing an evaluation mechanism of ADS development on both holistic and modular levels. In this work, we specifically tuned our experiments in a way that shows the feature development of the motion prediction module and its tightly coupled predecessor in the ADS pipeline, the perception module. In C2, we not only address real-life scenarios using motion prediction but also discuss how module-level shortcomings can bleed into successor modules of a modular AD pipeline.

Finally, for [RQ3], In C3, we constructed and validated a novel motion prediction model on a public dataset, which not only gave us insights into developing motion prediction model but also landed us in competition with the current SOTA on a public leaderboard.

Overall, these contributions represent significant advancements in autonomous driving from a research and practical ADS development perspective, giving robust insights about how specific traffic scenarios can be solved by incorporating prediction knowledge into navigation decision-making and what short-comings are faced while incorporating classical and/or data-oriented models into practical autonomy pipelines. The collective findings emphasize that prediction's performance is highly dependent on application context (integrated or isolated) and that the improvements in isolation do not guarantee system level gains.

7.1. Future Work

This thesis offers a bright opportunity for new research that can be done on top of it. Following are some of the possible routes that can be taken as a follow-up of this work:

1. **Adaptability:** The work done in this thesis blends applied and academic research. While both research types ideally tend to solve real-life problems, the applicability of academic research (performed on clean preprocessed datasets) in real-world comes with many challenges. One such challenge is domain adaptation i.e. models trained on a particular data could be prone to distribution shifts when tested on real-life data. Moreover, as discussed in Chapter 5, adapting any prediction model to practical autonomy pipelines comes with a dynamics gap. Working on reducing dynamics gap in motion prediction is one of the possible extensions of this thesis.
2. **Prediction Bottlenecks:** Motion prediction inherently suffers from pipeline bottlenecks, i.e., shortcomings of predecessor modules of autonomy are cascaded into the prediction module, as discussed in Chapter 4. Identifying and rectifying such bottlenecks is crucial and requires an in-depth analysis of how much of the performance of the prediction module is generally impacted due to external module factors. This problem is often overlooked because it is not applicable while working with motion prediction datasets but it has a visible impact on overall performance of the autonomy when working with autonomy stacks in general.
3. **Architectural Designs:** A classical follow-up of this work is to further improve the modeling architectures. Every day hundreds of SOTA models are introduced in research, which employ the latest advancements in deep learning-based model design. Here, a good follow-up would be to start with a goal-based prediction model and refine it such that the short-comings we experienced in Chapter-5 (illogical goals, wrong trajectories) can be addressed.

   In general, the performance of solving real-world scenarios using motion prediction can further be improved if one can either work on one or a combination of the above-mentioned prospective areas of research as a follow-up of this work.

BIBLIOGRAPHY

[1] World Health Organization. *Road Traffic Injuries*. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries. [Accessed 2025-06-12]. 2023.

[2] International Transport Forum. *Road Safety Annual Report 2023*. https://www.itf-oecd.org/road-safety-annual-report-2023. [Accessed 2025-06-12]. 2023.

[3] National Highway Traffic Safety Administration. *Research | NHTSA*. https://www.nhtsa.gov/research. [Accessed 2025-06-12]. 2025.

[4] SAE International. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles (J3016_202104)*. https://www.sae.org/standards/content/j3016_202104/. [Accessed 2025-06-12]. 2021.

[5] Ardi Tampuu et al. "A Survey of End-to-End Driving: Architectures and Training Methods". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.4 (2020), pp. 1364–1384.

[6] Chenyi Chen et al. "Deepdriving: Learning affordance for direct perception in autonomous driving". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2722–2730.

[7] Julian F. P. Kooij et al. "Context-Based Path Prediction for Targets with Switching Dynamics". In: *International Journal of Computer Vision* 127.3 (2019), pp. 239–262.

[8] Miaomiao Liu et al. "Analyzing Drivers' Crossing Decisions at Unsignalized Intersections in China". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 24 (2014), pp. 244–255.

[9] Naveed Muhammad. *Motion Prediction in Autonomous Driving*. Accessed: 2025-09-15. 2025. URL: https://www.youtube.com/watch?v=hTyz24C_GqY.

[10] Rodney Brooks. *The Big Problem With Self-Driving Cars Is People*. Accessed: 2025-06-12. 2023. URL: https://spectrum.ieee.org/the-big-problem-with-selfdriving-cars-is-people.

[11] Nachiket Deo, Eric Wolff, and Oscar Beijbom. "Multimodal Trajectory Prediction Conditioned on Lane-Graph Traversals". In: *Conference on Robot Learning*. PMLR. 2022, pp. 203–212.

[12] Sebastian Thrun et al. "Stanley: The Robot That Won the DARPA Grand Challenge". In: *Journal of Field Robotics* 23.9 (2006), pp. 661–692.

[13] Defense Advanced Research Projects Agency. *DARPA Grand Challenge 2005: Final Report*. Final Report. Available online. Defense Advanced Research Projects Agency (DARPA), Oct. 2005.

[14] Chris Urmson et al. "Autonomous Driving in Urban Environments: Boss and the Urban Challenge". In: *Journal of Field Robotics* 25.8 (2008), pp. 425–466.

[15] Defense Advanced Research Projects Agency. *DARPA Urban Challenge 2007: Final Report*. Final Report. Retrieved from DARPA website. Arlington, VA: Defense Advanced Research Projects Agency (DARPA), Nov. 2007.

[16] Shinpei Kato et al. "An Open Approach to Autonomous Vehicles". In: *IEEE Micro* 35.6 (2015), pp. 60–68. DOI: 10.1109/MM.2015.133.

[17] Baidu Apollo. *Apollo: An Open Autonomous Driving Platform*. https://github.com/ApolloAuto/apollo. Accessed: 2025-06-12. 2017.

[18] Wenyuan Zeng et al. “End-to-End Interpretable Neural Motion Planner”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8660–8669.

[19] Yi Xiao et al. “Multimodal End-to-End Autonomous Driving”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.1 (2020), pp. 537–547.

[20] Brenna D. Argall et al. “A Survey of Robot Learning from Demonstration”. In: *Robotics and Autonomous Systems* 57.5 (2009), pp. 469–483.

[21] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Vol. 1. 1. MIT Press Cambridge, 1998.

[22] Holger Caesar et al. “nuScenes: A Multimodal Dataset for Autonomous Driving”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11621–11631.

[23] Ming-Fang Chang et al. “Argoverse: 3D Tracking and Forecasting with Rich Maps”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8748–8757.

[24] Pei Sun et al. “Scalability in Perception for Autonomous Driving: Waymo Open Dataset”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2446–2454.

[25] R. Kesten et al. *Lyft Level 5 AV Dataset 2019*. https://level5.lyft.com/dataset. Accessed: 2025-06-12. 2019.

[26] Mahir Gulzar, Yar Muhammad, and Naveed Muhammad. “A Survey on Motion Prediction of Pedestrians and Vehicles for Autonomous Driving”. In: *IEEE Access* 9 (2021). Open access, pp. 137957–137969. DOI: 10.1109/ACCESS.2021.3118224.

[27] World Health Organization. *Global Status Report on Road Safety 2018*. World Health Organization, 2019.

[28] Matt McFarland. *Robots Hit the Streets. The Streets Hit Back*. http://money.cnn.com/2017/04/28/technology/robot-bullying/. Accessed: 2025-06-12. 2017.

[29] Sean Hollister. “I rode in a Hyundai Ioniq5 with wheels that go sideways”. In: *The Verge* (Jan. 2024). Accessed: 2025-06-12. URL: https://www.theverge.com/24035050/hyundai-mobis-e-corner-system-ioniq-5-crab-drive-360.

[30] Simon Ulbrich et al. “Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving”. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE. 2015, pp. 982–988.

[31] Gerrit Bagschik, Till Menzel, and Markus Maurer. “Ontology based scene creation for the development of automated vehicles”. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2018, pp. 1813–1820.

[32] Nachiket Deo and Mohan M Trivedi. “Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms”. In: *2018 IEEE intelligent vehicles symposium (IV)*. IEEE. 2018, pp. 1179–1184.

[33] Yongjun Yan et al. “A Multi-Vehicle Game-Theoretic Framework for Decision Making and Planning of Autonomous Vehicles in Mixed Traffic”. In: *IEEE Transactions on Intelligent Vehicles* 8.11 (2023), pp. 4572–4587. DOI: 10.1109/TIV.2023.3321346.

[34] Jaime F. Fisac et al. “Hierarchical Game-Theoretic Planning for Autonomous Vehicles”. In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 9590–9596. DOI: 10.1109/ICRA.2019.8794007.

[35] Meng Wang et al. “Game theoretic approach for predictive lane-changing and car-following control”. In: *Transportation Research Part C: Emerging Technologies* 58 (2015), pp. 73–92.

[36] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. “Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning”. In: *2019 international conference on robotics and automation (ICRA)*. IEEE. 2019, pp. 6658–6664.

[37] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. “Understanding pedestrian behavior in complex traffic scenes”. In: *IEEE Transactions on Intelligent Vehicles* 3.1 (2017), pp. 61–70.

[38] Amir Rasouli and John K Tsotsos. “Autonomous vehicles that interact with pedestrians: A survey of theory and practice”. In: *IEEE transactions on intelligent transportation systems* 21.3 (2019), pp. 900–918.

[39] Nina Brouwer, Horst Kloeden, and Christoph Stiller. “Comparison and evaluation of pedestrian motion models for vehicle safety systems”. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2016, pp. 2207–2212.

[40] Daniela Ridel et al. “A literature review on the prediction of pedestrian behavior in urban scenarios”. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 3105–3112.

[41] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. “A survey on motion prediction and risk assessment for intelligent vehicles”. In: *ROBOMECH journal* 1 (2014), pp. 1–14.

[42] Sajjad Mozaffari et al. “Deep learning-based vehicle behavior prediction for autonomous driving applications: A review”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.1 (2020), pp. 33–47.

[43] Andrey Rudenko et al. “Human motion trajectory prediction: A survey”. In: *The International Journal of Robotics Research* 39.8 (2020), pp. 895–935.

[44] Ivo Batkovic et al. “A computationally efficient model for pedestrian motion prediction”. In: *2018 European control conference (ECC)*. IEEE. 2018, pp. 374–379.

[45] Andreas Møgelmose, Mohan M Trivedi, and Thomas B Moeslund. “Trajectory analysis and prediction for improved pedestrian safety: An integrated framework and evaluations”. In: *2015 IEEE intelligent vehicles symposium (IV)*. IEEE. 2015, pp. 330–335.

[46] Nico Kaempchen et al. “IMM object tracking for high dynamic driving maneuvers”. In: *IEEE Intelligent Vehicles Symposium, 2004*. IEEE. 2004, pp. 825–830.

[47] Panagiotis Lytrivis, George Thomaidis, and Angelos Amditis. “Cooperative path prediction in vehicular environments”. In: *2008 11th international IEEE conference on intelligent transportation systems*. IEEE. 2008, pp. 803–808.

[48] Guotao Xie et al. “Vehicle trajectory prediction by integrating physics-and maneuver-based approaches using interactive multiple models”. In: *IEEE Transactions on Industrial Electronics* 65.7 (2017), pp. 5999–6008.

[49] Lamberto Ballan et al. “Knowledge transfer for scene-specific motion prediction”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The*

Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer. 2016, pp. 697–713.

[50] Tobias Gindele, Sebastian Brechtel, and Rüdiger Dillmann. "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments". In: *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE. 2010, pp. 1625–1631.

[51] Federico Bartoli et al. "Context-aware trajectory prediction". In: *2018 24th international conference on pattern recognition (ICPR)*. IEEE. 2018, pp. 1941–1946.

[52] Hao Xue, Du Q Huynh, and Mark Reynolds. "SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction". In: *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 2018, pp. 1186–1194.

[53] Shengzhe Dai, Li Li, and Zhiheng Li. "Modeling vehicle interactions via modified LSTM models for trajectory prediction". In: *Ieee Access* 7 (2019), pp. 38287–38296.

[54] Nachiket Deo and Mohan M Trivedi. "Convolutional social pooling for vehicle trajectory prediction". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 1468–1476.

[55] Tianyang Zhao et al. "Multi-agent tensor fusion for contextual trajectory prediction". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 12126–12134.

[56] Agrim Gupta et al. "Social gan: Socially acceptable trajectories with generative adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2255–2264.

[57] Li-Wei Kang et al. "Vehicle trajectory prediction based on social generative adversarial network for self-driving car applications". In: *2020 international symposium on computer, consumer and control (IS3C)*. IEEE. 2020, pp. 489–492.

[58] Nachiket Deo, Eric Wolff, and Oscar Beijbom. "Multimodal Trajectory Prediction Conditioned on Lane-Graph Traversals". In: *Conference on Robot Learning*. PMLR. 2022, pp. 203–212.

[59] Aniket Bera et al. "GLMP-realtime pedestrian path prediction using global and local movement patterns". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 5528–5535.

[60] Stefan Atev, Grant Miller, and Nikolaos P Papanikolopoulos. "Clustering of vehicle trajectories". In: *IEEE transactions on intelligent transportation systems* 11.3 (2010), pp. 647–657.

[61] Fazeleh S Hoseini, Sadegh Rahrovani, and Morteza Haghir Chehreghani. "A generic framework for clustering vehicle trajectories". In: *arXiv preprint arXiv:2009.12443* (2020).

[62] John Martinsson, Nasser Mohammadiha, and Alexander Schliep. "Clustering vehicle maneuver trajectories using mixtures of hidden markov models". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 3698–3705.

[63] Nishant Nikhil and Brendan Tran Morris. "Convolutional neural network for trajectory prediction". In: *Proceedings of the European conference on computer vision (ECCV) workshops*. 2018, pp. 0–0.

[64] Oily Styles, Arun Ross, and Victor Sanchez. "Forecasting pedestrian trajectory with machine-annotated training data". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 716–721.

[65] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[66] Shashank Srikanth et al. “Infer: Intermediate representations for future prediction”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 942–949.

[67] Donghan Lee et al. “Convolution neural network-based lane change intention prediction of surrounding vehicles for ACC”. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2017, pp. 1–6.

[68] Sergio Casas, Wenjie Luo, and Raquel Urtasun. “Intentnet: Learning to predict intention from raw sensor data”. In: *Conference on Robot Learning*. PMLR. 2018, pp. 947–956.

[69] Alexandre Alahi et al. “Social lstm: Human trajectory prediction in crowded spaces”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 961–971.

[70] Fang-Chieh Chou et al. “Predicting motion of vulnerable road users using high-definition maps and efficient convnets”. In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, pp. 1655–1662.

[71] Nemanja Djuric et al. “Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 2095–2104.

[72] United Nations Economic Commission for Europe. *Convention on Road Signs and Signals*. `https://unece.org/DAM/trans/conventn/Conv_road_signs_2006v_EN.pdf`. Accessed: Oct. 29, 2022. 2006.

[73] Hatem Darweesh et al. “Open source integrated planner for autonomous navigation in highly dynamic environments”. In: *Journal of Robotics and Mechatronics* 29.4 (2017), pp. 668–684.

[74] CARLA Team. *CARLA NHTSA Scenarios*. `https://carlachallenge.org/challenge/nhtsa/`. Accessed: Oct. 29, 2022.

[75] UT-ADL. *autoware_mini*. `https://github.com/UT-ADL/autoware_mini`. Accessed: Jun. 12, 2025. 2023.

[76] Nguyen Mau Dung. *Super-Fast-Accurate-3D-Object-Detection-PyTorch*. `https://github.com/maudzung/Super-Fast-Accurate-3D-Object-Detection`. 2020.

[77] Henggang Cui et al. “Multimodal trajectory predictions for autonomous driving using deep convolutional networks”. In: *2019 international conference on robotics and automation (icra)*. IEEE. 2019, pp. 2090–2096.

[78] Tim Salzmann et al. “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer. 2020, pp. 683–700.

[79] Tung Phan-Minh et al. “Covernet: Multimodal behavior prediction using trajectory sets”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 14074–14083.

[80] Jiyang Gao et al. “Vectornet: Encoding hd maps and agent dynamics from vectorized representation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11525–11533.

[81] Jingke Wang et al. “Ltp: Lane-based trajectory prediction for autonomous driving”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 17134–17142.

[82] Mengmeng Liu et al. “Laformer: Trajectory prediction for autonomous driving with lane-aware scene constraints”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 2039–2049.

[83] Sandra Carrasco Limeros et al. “Towards explainable motion prediction using heterogeneous graph representations”. In: *Transportation Research Part C: Emerging Technologies* 157 (2023), p. 104405.

[84] Jiuyu Chen et al. “Q-EANet: Implicit social modeling for trajectory prediction via experience-anchored queries”. In: *IET Intelligent Transport Systems* 18.6 (2024), pp. 1004–1015.

[85] Yuning Chai et al. “Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction”. In: *arXiv preprint arXiv:1910.05449* (2019).

[86] Ye Yuan et al. “Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9813–9823.

[87] Kaouther Messaoud et al. “Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation”. In: *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2021, pp. 165–170.

[88] Chenxu Luo et al. “Probabilistic multi-modal trajectory prediction with lane attention for autonomous vehicles”. In: *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2020, pp. 2370–2376.

[89] ByeoungDo Kim et al. “Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14636–14645.

[90] Thomas Gilles et al. “Gohome: Graph-oriented heatmap output for future motion estimation”. In: *2022 international conference on robotics and automation (ICRA)*. IEEE. 2022, pp. 9107–9114.

[91] Roger Girgis et al. “Latent variable sequential set transformers for joint multi-agent motion prediction”. In: *arXiv preprint arXiv:2104.00563* (2021).

[92] Thomas Gilles et al. “Thomas: Trajectory heatmap output with learned multi-agent sampling”. In: *arXiv preprint arXiv:2110.06607* (2021).

[93] Chen Feng et al. “Macformer: Map-agent coupled transformer for real-time and robust trajectory prediction”. In: *IEEE Robotics and Automation Letters* (2023).

[94] Phong Tran et al. “What truly matters in trajectory prediction for autonomous driving?” In: *arXiv preprint arXiv:2306.15136* (2023).

ACKNOWLEDGEMENTS

First of all, I would like to thank the Almighty Allah SWT for giving me the opportunity to pursue and finish this doctoral degree.

I would like to extend sincere gratitude to my supervisor, Naveed Muhammad, for his support and mentorship during my academic journey. His guidance and patience have been crucial in my development as a researcher in robotics. I am grateful for the research insights he has imparted to me. His dedication to my growth as a professional has been truly commendable.

I would like to thank my co-supervisor, Yar Muhammad, for his support and encouragement throughout my academic journey.

I wish to express my heartfelt appreciation to my work supervisor, Tambet Matiisen. He has been a guiding force throughout my journey into the autonomous driving industry. His exceptional programming and management skills have taught me a lot. Whenever I faced challenges during my doctoral work, he was always there with a helping hand and invaluable support.

I would like to lovingly remember my late father, Gulzar Hussain Abid, and express my deepest gratitude to my mother, Asifa Sultana. Their unconditional love, sacrifices, and prayers have been a constant source of strength and motivation throughout this journey.

My deepest gratitude goes to my dearest wife and best friend, Fareeha. Her unwavering love, patience, and belief in me have been the foundation of my perseverance. Through every challenge and success, she stood by my side with remarkable strength and grace, making this journey possible.

I am profoundly grateful to Sadi Alawadi and Eren Erdal Aksoy for serving as the reviewers and the opponents of this thesis.

Lastly, I would like to acknowledge my friends and colleagues Mubashir, Faiz, Edgar, Kertu, Zain, Allan, Laima, Dymtro, Ardi, and Debasis. Their substantial assistance played an important role in completing this work.

SISUKOKKUVÕTE

Reaalsete liiklusolukordade lahendamine isejuhtivates autodes liikumise ennustamise abil

Isejuhtivate sõidukite valdkond on viimase kümne aasta jooksul kiiresti arenenud. Sellele on oluliselt kaasa aidanud tehisintellekti plahvatuslik areng. Transpordiettevõtted üle maailma võistlevad omavahel, et jõuda esimestena osalise või täieliku autonoomsuse rakendamiseni. Et uusimad teadustulemused jõuaksid kiiresti praktilisse kasutusse, on võtmetähtsusega tihe koostöö ettevõtete ja akadeemia vahel.

Kaks peamist lähenemist autonoomse sõiduki juhtimisele on modulaarne lahendus ja närvivõrkudel põhinev (end-to-end) lähenemine. Praktikas eelistatakse tihtipeale modulaarset lahendust, sest see jagab keerulise isejuhtimise ülesande väiksemateks mooduliteks. Valitud lahendusest sõltumata on üheks võtmetähtsusega komponendiks isejuhtivuse juures teiste liiklejate käitumise ennustamine. Edukaks navigeerimiseks keerulistes liiklusolukordades peab isejuhtiv sõiduk suutma ennustada teiste liiklejate tõenäolisi liikumistrajektoore, mis omakorda on aluseks takistusteta ja ohutu trajektoori valimiseks sõidukile endale. Antud töös keskendutakse sellele, kuidas isejuhtiva auto käitumist keerukates liiklusolukordades paremini mõista ning paremaks teha läbi teiste liiklejate käitumise parema ennustamise.

Alustuseks tehakse põhjalik kirjanduse ülevaade erinevatest liikumise ennustamise modelleerimise meetoditest, et mõista erinevate mudelite tugevusi ja nõrkusi enne nende rakendamist konkreetsetes liiklusstsenaariumites. Pakutakse välja taksonoomia, mis jaotab ennustusmudeleid lähtuvalt kasutatavast meetodist, väljundist ning olukorrateadlikkusest. See taksonoomia toimib edasise uurimistöö alusena.

Järgnevas osas rakendatakse erinevaid liiklejate käitumise ennustamise meetodeid, et lahendada isejuhtiva auto käitumist keerukates liiklussituatsioonides. Näiteks pakutakse välja vabavaraline lahendus “anna teed” olukordade lahendamiseks ristmikel, mis kasutab mitmemodaalseid trajektooriennustusi teiste liiklejate kohta. See uurimus näitas ühtlasi, kui oluline on ennustusele eelnevate moodulite sisendi kvaliteet, kuna nende puudujäägid kanduvad edasi ka ennustusmooduli tulemustesse.

Seejärel vaadeldakse käitumise ennustamise mõju isejuhtiva süsteemi toimimisele laiemalt. Välja pakutakse stsenaariumpõhine hindamisraamistik isejuhtivuse lahenduste hindamiseks nii mooduli kui terviku tasemel. See annab sügavama arusaama moodulitevahelistest sõltuvustest ja nende mõjust erinevates liiklusolukordades.

Viimases osas luuakse uus liikumise ennustusmudel, mis saavutab konkurentsivõimelisi tulemusi rahvusvahelises võrdlustestis. Pakutud mudel kasutab eesmärgipõhist tingimist sõiduradade graafi peal ning väljastab mitmemodaalseid trajektoore. Lisaks arutletakse praktiliste väljakutsete üle, mis kaasnevad tipptasemel teadustulemuste integreerimisega olemasolevatesse isejuhtivuse lahendustesse.

Kokkuvõttes ühendab töö edukalt nii akadeemilise kui ka praktilise uurimistöö. See annab olulise panuse autonoomsete sõidukite arengusse, pakkudes paremat arusaama, kuidas liiklusolukordi saab lahendada liikumisennustuse abil ning kuidas erinevad autonoomia moodulid – eriti just ennustusmoodul – omavahel tihedalt seotud on.

PUBLICATIONS

CURRICULUM VITAE

Personal data

Name:	Mahir Gulzar
Date of birth:	06.06.1993
Citizenship:	Pakistan
Contact:	mahir.gulzar@ut.ee

Education

2020–2026	PhD in Computer Science, University of Tartu, Estonia
2017–2019	MSc in Computer Science, University of Tartu, Estonia
2012–2016	BS in Computer Science, Government College University Lahore, Pakistan

Employment

2024–Present	Senior Software Engineer, Starship Technologies OÜ, Tallinn, Estonia
2020–2024	Autonomy Engineer & Junior Research Fellow of Autonomous Driving, Institute of Computer Science, University of Tartu, Tartu, Estonia
2019–2020	Scientific Programmer, Institute of Computer Science, University of Tartu, Tartu, Estonia
2018	SpaceTEM Intern, Milrem Robotics, Tartu, Estonia
2016–2017	Software Engineer (Unity Programmer), OziTechnology, Lahore, Pakistan

Scientific work

Main fields of interest:

- Perception & Computer Vision
- Behaviour Prediction
- Tracking
- Motion Planning
- Simulations

ELULOOKIRJELDUS

Isikuandmed

Nimi:	Mahir Gulzar
Sünnikuupäev:	06.06.1993
Kodakondsus:	Pakistan
Kontakt:	mahir.gulzar@ut.ee

Haridus

2020–2026	Doktorikraad arvutiteaduses, Tartu Ülikool, Eesti
2017–2019	Magistrikraad arvutiteaduses, Tartu Ülikool, Eesti
2012–2016	Bakalaureusekraad arvutiteaduses, Government College University Lahore, Pakistan

Teenistuskäik

2024–Praeguseni	Vanem tarkvarainsener, Starship Technologies OÜ, Tallinn, Eesti
2020–2024	Autonoomsuse insener ja nooremteadur, Tartu Ülikooli arvutiteaduse instituut, Tartu, Eesti
2019–2020	Teadusprogrammeerija, Tartu Ülikooli arvutiteaduse instituut, Tartu, Eesti
2018	SpaceTEM praktikant, Milrem Robotics, Tartu, Eesti
2016–2017	Tarkvarainsener (Unity programmeerija), OziTechnology, Lahore, Pakistan

Teadustegevus

Peamised uurimisvaldkonnad:

- Tajumine ja arvutinägemine
- Käitumise ennustamine
- Objektijälgimine
- Liikumise planeerimine
- Simulatsioonid

DISSERTATIONES INFORMATICAE PREVIOUSLY PUBLISHED IN DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS

19. **Helger Lipmaa.** Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
22. **Kaili Müürisep.** Eesti keele arvutigrammatika: süntaks. Tartu, 2000, 107 lk.
23. **Varmo Vene.** Categorical programming with inductive and coinductive types. Tartu, 2000, 116 p.
24. **Olga Sokratova.** Ω-rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
27. **Tiina Puolakainen.** Eesti keele arvutigrammatika: morfoloogiline ühestamine. Tartu, 2001, 138 lk.
29. **Jan Villemson.** Size-efficient interval time stamps. Tartu, 2002, 82 p.
45. **Kristo Heero.** Path planning and learning strategies for mobile robots in dynamic partially unknown environments. Tartu 2006, 123 p.
49. **Härmel Nestra.** Iteratively defined transfinite trace semantics and program slicing with respect to them. Tartu 2006, 116 p.
53. **Marina Issakova.** Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment. Tartu 2007, 170 p.
55. **Kaarel Kaljurand.** Attempto controlled English as a Semantic Web language. Tartu 2007, 162 p.
56. **Mart Anton.** Mechanical modeling of IPMC actuators at large deformations. Tartu 2008, 123 p.
59. **Reimo Palm.** Numerical Comparison of Regularization Algorithms for Solving Ill-Posed Problems. Tartu 2010, 105 p.
61. **Jüri Reimand.** Functional analysis of gene lists, networks and regulatory systems. Tartu 2010, 153 p.
62. **Ahti Peder.** Superpositional Graphs and Finding the Description of Structure by Counting Method. Tartu 2010, 87 p.
64. **Vesal Vojdani.** Static Data Race Analysis of Heap-Manipulating C Programs. Tartu 2010, 137 p.
66. **Mark Fišel.** Optimizing Statistical Machine Translation via Input Modification. Tartu 2011, 104 p.
67. **Margus Niitsoo**. Black-box Oracle Separation Techniques with Applications in Time-stamping. Tartu 2011, 174 p.
71. **Siim Karus.** Maintainability of XML Transformations. Tartu 2011, 142 p.
72. **Margus Treumuth.** A Framework for Asynchronous Dialogue Systems: Concepts, Issues and Design Aspects. Tartu 2011, 95 p.
73. **Dmitri Lepp.** Solving simplification problems in the domain of exponents, monomials and polynomials in interactive learning environment T-algebra. Tartu 2011, 202 p.

74. **Meelis Kull.** Statistical enrichment analysis in algorithms for studying gene regulation. Tartu 2011, 151 p.
77. **Bingsheng Zhang.** Efficient cryptographic protocols for secure and private remote databases. Tartu 2011, 206 p.
78. **Reina Uba.** Merging business process models. Tartu 2011, 166 p.
79. **Uuno Puus.** Structural performance as a success factor in software development projects – Estonian experience. Tartu 2012, 106 p.
81. **Georg Singer.** Web search engines and complex information needs. Tartu 2012, 218 p.
83. **Dan Bogdanov.** Sharemind: programmable secure computations with practical applications. Tartu 2013, 191 p.
84. **Jevgeni Kabanov.** Towards a more productive Java EE ecosystem. Tartu 2013, 151 p.
87. **Margus Freudenthal.** Simpl: A toolkit for Domain-Specific Language development in enterprise information systems. Tartu, 2013, 151 p.
90. **Raivo Kolde.** Methods for re-using public gene expression data. Tartu, 2014, 121 p.
91. **Vladimir Šor.** Statistical Approach for Memory Leak Detection in Java Applications. Tartu, 2014, 155 p.
92. **Naved Ahmed.** Deriving Security Requirements from Business Process Models. Tartu, 2014, 171 p.
94. **Liina Kamm.** Privacy-preserving statistical analysis using secure multi-party computation. Tartu, 2015, 201 p.
100. **Abel Armas Cervantes.** Diagnosing Behavioral Differences between Business Process Models. Tartu, 2015, 193 p.
101. **Fredrik Milani.** On Sub-Processes, Process Variation and their Interplay: An Integrated Divide-and-Conquer Method for Modeling Business Processes with Variation. Tartu, 2015, 164 p.
102. **Huber Raul Flores Macario.** Service-Oriented and Evidence-aware Mobile Cloud Computing. Tartu, 2015, 163 p.
103. **Tauno Metsalu.** Statistical analysis of multivariate data in bioinformatics. Tartu, 2016, 197 p.
104. **Riivo Talviste.** Applying Secure Multi-party Computation in Practice. Tartu, 2016, 144 p.
108. **Siim Orasmaa.** Explorations of the Problem of Broad-coverage and General Domain Event Analysis: The Estonian Experience. Tartu, 2016, 186 p.
109. **Prastudy Mungkas Fauzi.** Efficient Non-interactive Zero-knowledge Protocols in the CRS Model. Tartu, 2017, 193 p.
110. **Pelle Jakovits.** Adapting Scientific Computing Algorithms to Distributed Computing Frameworks. Tartu, 2017, 168 p.
111. **Anna Leontjeva.** Using Generative Models to Combine Static and Sequential Features for Classification. Tartu, 2017, 167 p.
112. **Mozhgan Pourmoradnasseri.** Some Problems Related to Extensions of Polytopes. Tartu, 2017, 168 p.

113. **Jaak Randmets.** Programming Languages for Secure Multi-party Computation Application Development. Tartu, 2017, 172 p.
114. **Alisa Pankova.** Efficient Multiparty Computation Secure against Covert and Active Adversaries. Tartu, 2017, 316 p.
116. **Toomas Saarsen.** On the Structure and Use of Process Models and Their Interplay. Tartu, 2017, 123 p.
121. **Kristjan Korjus.** Analyzing EEG Data and Improving Data Partitioning for Machine Learning Algorithms. Tartu, 2017, 106 p.
122. **Eno Tõnisson.** Differences between Expected Answers and the Answers Offered by Computer Algebra Systems to School Mathematics Equations. Tartu, 2017, 195 p.

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

1. **Abdullah Makkeh**. Applications of Optimization in Some Complex Systems. Tartu 2018, 179 p.
2. **Riivo Kikas**. Analysis of Issue and Dependency Management in Open-Source Software Projects. Tartu 2018, 115 p.
3. **Ehsan Ebrahimi**. Post-Quantum Security in the Presence of Superposition Queries. Tartu 2018, 200 p.
4. **Ilya Verenich**. Explainable Predictive Monitoring of Temporal Measures of Business Processes. Tartu 2019, 151 p.
5. **Yauhen Yakimenka**. Failure Structures of Message-Passing Algorithms in Erasure Decoding and Compressed Sensing. Tartu 2019, 134 p.
6. **Irene Teinemaa**. Predictive and Prescriptive Monitoring of Business Process Outcomes. Tartu 2019, 196 p.
7. **Mohan Liyanage.** A Framework for Mobile Web of Things. Tartu 2019, 131 p.
8. **Toomas Krips.** Improving performance of secure real-number operations. Tartu 2019, 146 p.
9. **Vijayachitra Modhukur.** Profiling of DNA methylation patterns as biomarkers of human disease. Tartu 2019, 134 p.
10. **Elena Sügis.** Integration Methods for Heterogeneous Biological Data. Tartu 2019, 250 p.
11. **Tõnis Tasa.** Bioinformatics Approaches in Personalised Pharmacotherapy. Tartu 2019, 150 p.
12. **Sulev Reisberg.** Developing Computational Solutions for Personalized Medicine. Tartu 2019, 126 p.
13. **Huishi Yin.** Using a Kano-like Model to Facilitate Open Innovation in Requirements Engineering. Tartu 2019, 129 p.
14. **Faiz Ali Shah.** Extracting Information from App Reviews to Facilitate Software Development Activities. Tartu 2020, 149 p.
15. **Adriano Augusto**. Accurate and Efficient Discovery of Process Models from Event Logs. Tartu 2020, 194 p.
16. **Karim Baghery.** Reducing Trust and Improving Security in zk-SNARKs and Commitments. Tartu 2020, 245 p.
17. **Behzad Abdolmaleki.** On Succinct Non-Interactive Zero-Knowledge Protocols Under Weaker Trust Assumptions. Tartu 2020, 209 p.
18. **Janno Siim.** Non-Interactive Shuffle Arguments. Tartu 2020, 154 p.
19. **Ilya Kuzovkin.** Understanding Information Processing in Human Brain by Interpreting Machine Learning Models. Tartu 2020, 149 p.
20. **Orlenys López Pintado.** Collaborative Business Process Execution on the Blockchain: The Caterpillar System. Tartu 2020, 170 p.
21. **Ardi Tampuu.** Neural Networks for Analyzing Biological Data. Tartu 2020, 152 p.

22. **Madis Vasser.** Testing a Computational Theory of Brain Functioning with Virtual Reality. Tartu 2020, 106 p.
23. **Ljubov Jaanuska.** Haar Wavelet Method for Vibration Analysis of Beams and Parameter Quantification. Tartu 2021, 192 p.
24. **Arnis Parsovs.** Estonian Electronic Identity Card and its Security Challenges. Tartu 2021, 214 p.
25. **Kaido Lepik.** Inferring causality between transcriptome and complex traits. Tartu 2021, 224 p.
26. **Tauno Palts.** A Model for Assessing Computational Thinking Skills. Tartu 2021, 134 p.
27. **Liis Kolberg.** Developing and applying bioinformatics tools for gene expression data interpretation. Tartu 2021, 195 p.
28. **Dmytro Fishman.** Developing a data analysis pipeline for automated protein profiling in immunology. Tartu 2021, 155 p.
29. **Ivo Kubjas.** Algebraic Approaches to Problems Arising in Decentralized Systems. Tartu 2021, 120 p.
30. **Hina Anwar.** Towards Greener Software Engineering Using Software Analytics. Tartu 2021, 186 p.
31. **Veronika Plotnikova.** FIN-DM: A Data Mining Process for the Financial Services. Tartu 2021, 197 p.
32. **Manuel Camargo.** Automated Discovery of Business Process Simulation Models From Event Logs: A Hybrid Process Mining and Deep Learning Approach. Tartu 2021, 130 p.
33. **Volodymyr Leno.** Robotic Process Mining: Accelerating the Adoption of Robotic Process Automation. Tartu 2021, 119 p.
34. **Kristjan Krips.** Privacy and Coercion-Resistance in Voting. Tartu 2022, 173 p.
35. **Elizaveta Yankovskaya.** Quality Estimation through Attention. Tartu 2022, 115 p.
36. **Mubashar Iqbal.** Reference Framework for Managing Security Risks Using Blockchain. Tartu 2022, 203 p.
37. **Jakob Mass.** Process Management for Internet of Mobile Things. Tartu 2022, 151 p.
38. **Gamal Elkoumy.** Privacy-Enhancing Technologies for Business Process Mining. Tartu 2022, 135 p.
39. **Lidia Feklistova.** Learners of an Introductory Programming MOOC: Background Variables, Engagement Patterns and Performance. Tartu 2022, 151 p.
40. **Mohamed Ragab.** Bench-Ranking: A Prescriptive Analysis Approach for Large Knowledge Graphs Query Workloads. Tartu 2022, 158 p.
41. **Mohammad Anagreh.** Privacy-Preserving Parallel Computations for Graph Problems. Tartu 2023, 181 p.
42. **Rahul Goel.** Mining Social Well-being Using Mobile Data. Tartu 2023, 104 p.

43. **Anti Ingel.** Algorithms using information theory: classification in brain-computer interfaces and characterising reinforcement-learning agents. Tartu 2023, 142 p.
44. **Shakshi Sharma.** Fighting Misinformation in the Digital Age: A Comprehensive Strategy for Characterizing, Identifying, and Mitigating Misinformation on Online Social Media Platforms. Tartu 2023, 158 p.
45. **Kristiina Rahkema.** Quality Analysis of iOS Applications with Focus on Maintainability and Security Aspects. Tartu 2023, 182 p.
46. **Ivan Slobozhan.** Studying Online Social Media Engagement in CIS Countries during Protests, Mass Demonstrations and War. Tartu 2023, 81 p.
47. **Nurlan Kerimov.** Building a catalogue of molecular quantitative trait loci to interpret complex trait associations. Tartu 2023, 248 p.
48. **Pavlo Tertychnyi.** Machine Learning Methods for Anti-Money Laundering Monitoring. Tartu 2023, 117 p.
49. **Abasi-amefon Obot Affia.** A Framework and Teaching Approach for IoT Security Risk Management. Tartu 2023, 180 p.
50. **Raimond-Hendrik Tunnel.** Video Game Design and Development Bachelor's Curriculum for Estonia. Tartu 2024, 137 p.
51. **Ahto Salumets.** Bioinformatics analysis of various aspects in immunology. Tartu 2024, 198 p.
52. **Mohammed Abdulhameed Shaif Ali.** Deep Learning Methods for Cell Microscopy Image Analysis. Tartu 2024, 143 p.
53. **Pille Pullonen-Raudvere**. Foundations of Efficient and Secure Algorithm Development for Secure Multiparty Computation. Tartu 2024, 265 p.
54. **Marili Rõõm.** Multiple approaches to learners' success and factors affecting it in computer programming MOOCs. Tartu 2024, 170 p.
55. **Shivananda Rangappa Poojara.** Design and Orchestration of Scalable, Event-Driven Serverless Data Pipelines for Internet of Things (IoT) Applications. Tartu 2024, 172 p.
56. **Hassan Abdulgaleel Hassan Salim Eldeeb.** Empowering Machine Learning Pipelines with Automated Feature Engineering. Tartu 2024, 121 p.
57. **Muhammad Uzair.** Soft decision making for agri-food 4.0. Tartu 2024, 158 p.
58. **Kirill Milintsevich.** Estimation of Depression Level from Text: Symptom-Based Approach, External Knowledge, Dataset Validity. Tartu 2024, 130 p.
59. **Maksym Del.** Multilingual and Multi-Domain Representational Patterns Across Trpansformer-Based Models. Tartu 2024, 131 p.
60. **Kristo Raun.** Adaptive Out-of-order Handling in Streaming Conformance Checking. Tartu 2024, 118 p.
61. **Toivo Vajakas.** Towards integration of mobile network data into analyzing human mobility. Tartu 2024, 103 p.
62. **Katsiaryna Lashkevich.** Data-Driven Analysis and Optimization of Waiting Times in Business Processes. Tartu 2024, 169 p.
63. **Alejandra Duque-Torres.** Classifying, Constraining and Ranking Metamorphic Relations. Tartu 2025, 159 p.

64. **Mariia Bakhtina.** A Method for Information Security and Privacy Management in Smart Solutions. Tartu 2025, 199 p.
65. **Andre Tättar.** Multilingual Machine Translation for Under-Resourced Languages. Tartu 2025, 170 p.
66. **Mahmoud Shoush.** Prescriptive Process Monitoring Under Uncertainty and Resource Constraints. Tartu 2025, 178 p.
67. **Alireza Akhavi Zadegan.** A Multimodal approach for refining Mapping and Localization by Integrating Generative AI and Pedestrian-Centric Data. Tartu 2025, 147 p.
68. **Eerik Muuli.** Automating the assessment and feedback processes in IT teaching – improving creation and maintenance from the teaching staff perspective. Tartu 2025, 196 p.
69. **Kateryna Kubrak.** Towards User-Centered Prescriptive Process Monitoring Systems. Tartu 2025, 151 p.
70. **Zhigang Yin.** Computing and Sensing in a Smart Ring. Tartu 2025, 251 p.
71. **Abdul-Rasheed Olatunji Ottun.** Practical Trustworthy Artificial Intelligence with Human Oversight. Tartu 2025, 239 p.
72. **Sander Mikelsaar.** Analysis and Optimization of Iteratively Decodable Codes. Tartu 2025, 146 p.
73. **Marharyta Domnich.** Advancing Human-Centric Counterfactual Explanations in Explainable AI. Tartu 2025, 210 p.
74. **Viacheslav Komisarenko.** Aligning Training Loss to Evaluation Metrics in Deep Learning. Tartu 2026, 165 p.
75. **Heidi Taveter.** Using Programming-Process Data of Introductory Programming Courses: Finding Solver Types, Giving Feedback, and Detecting Plagiarism. Tartu 2026, 184 p.
76. **Daniel Majoral Lopez.** Deep neural networks for microscopy images. Tartu 2026, 81 p.