Universidad de Alcalá

PhD. Program in Electronics: Advanced Electronic Systems. Intelligent Systems

Predictive Techniques for Scene Understanding by using Deep Learning in Autonomous Driving

PhD. Thesis Presented by

Carlos Gómez Huélamo

2023

Universidad de Alcalá

PhD. Program in Electronics: Advanced Electronic Systems. Intelligent Systems

Predictive Techniques for Scene Understanding by using Deep Learning in Autonomous Driving

PhD. Thesis Presented by

Carlos Gómez Huélamo

Advisors

Dr. Luis Miguel Bergasa Pascual

Dr. Rafael Barea Navarro

Alcalá de Henares, TBD

A mi Madre, allá donde esté ......

“En este vasto mundo
navegáis en pos de un sueño,
surcando el ancho mar
que se extiende frente a vosotros.
El puerto de destino es el mañana
cada día más incierto.
Encontrad el camino,
cumplid vuestros sueños,
estáis todos en el mismo barco
y vuestra bandera es la libertad“
Opening 3 de One Piece
Autor: The Babystars

Acknowledgements

Esta Tesis Doctoral supone el culmen a cuatro años (Abril 2019 - Abril 2023) realmente duros, cargado de emociones, triunfos, pandemias, estafas y tropiezos, todo a partes iguales. Este es probablemente (aunque como diría Sean Connery interpretando a James Bond en 1983, *Never Say Never Again*) mi último gran documento individual, académicamente hablando.

Durante mi etapa universitaria (2013 hasta el momento, 2023) he tenido ciertos momentos puntuales en los que he sentido un salto cualitativo como profesional: El primero fue en el segundo cuatrimestre de segundo de carrera, cuando las cosas se pusieron tensas con Control II e Informática Industrial. Vaya sudores. El segundo probablemente fue con el fallecimiento de mi madre durante mi ERASMUS+ en Irlanda. Duros y oscuros momentos, alejado de mis seres queridos. El tercer momento llega en segundo de máster, durante mi querido ERASMUS+ en Finlandia, donde compagino una estancia preciosa en Tampere con el máster y un pre-inicio de doctorado. Me equivoqué al empezar tan pronto con la beca, "queriendo cobrar" cuanto antes, en vez de terminar tranquilamente el TFM y plantear la tesis, pero eso no lo sabría hasta tiempo después. Pero no es hasta la tesis donde empezaron los quebraderos de cabeza reales. Continuamente altibajos, mala planificación por mi parte, momentos puntuales donde me equivoqué rotundamente al empecinarme en soldar una estructura compleja para nuestro vehículo sin ayuda, no estudiar PyTorch tras el congreso WAF 2018 tras la sugerencia de mi tutor, no enfocarme en técnica individual hasta bien entrado el doctorado, no querer hacer nada hasta que no tuviese la teoría perfectamente asimilada, tener demasiado respeto a la Inteligencia Artificial y escurrir el bulto de mi tesis en un compañero mientras yo me dedicaba a integrar y corregir los bugs del grupo que para mí *era lo fácil*. Mal. Todo mal. Pero todo cambió tras mi segunda estancia, en Estados Unidos, cuando tras llorar por no entender el camino a seguir, nadie que me ayudara, decidí crear mi propio camino, con paciencia y fé, práctica y error compaginado con lectura de artículos, para mejorar mi confianza y autoestima, y finalmente logré empezar a entender lo que era el Deep Learning. Gracias a todos mis errores, desventuras y discusiones, a día de hoy, excepto momentos inevitables, me encuentro con muchísima capacidad para atacar y gestionar prácticamente cualquier problema, consultar documentación y organizarme, aunque esta sigue siendo mi tarea pendiente.

Cada año, desde hace ya varios, mi primera publicación en Instagram viene seguida de la frase "Trabaja duro en silencio y deja que tu éxito haga todo el ruido". Filosofía Kaizen, de mejora y aprendizaje continuo, para así cada día entender el mundo un poquito mejor. Si toda la dedicación y estudio que he depositado en este trabajo sirven para algo en mi futuro, sé que todo el esfuerzo habrá merecido la pena.

Después de este particular monólogo, a lo cual soy muy propenso y de lo cual mis amigos y compañeros no cesan en su empeño de recordádmelo, debo, como no puede ser de otra manera, dar paso a los agradecimientos.

En primer lugar, me gustaría agradecer a mis profesores del grupo RobeSafe, especialmente a mis tutores Luis Miguel Bergasa Pascual y Rafael Barea Navarro, por ofrecerme estar en el grupo (así como aguantarme) durante todos estos años e intentar que tuviésemos el mejor *experimental setup* y *roadmap* en el laboratorio, aunque no fuese siempre sencillo. Sin dudas considero realmente interesante la temática propuesta en esta tesis doctoral, predicción de agentes en el contexto de vehículo inteligente, ya que entra en el plano filosófico sobre cómo razonar el futuro de los objetos y cómo podría afectar a la capacidad ejecutiva del agente que deba tomar una decisión. Mi mente hace tiempo que cambió y me fijo siempre que conduzco de todo lo que intento reproducir con mis estudios. Habrá que seguir esta tendencia muy de cerca en los próximos años, porque personalmente considero que sus aplicaciones son fantásticas.

A mis tutores en las estancias de doctorado, Christoph Stiller y Eduardo Molinos en el Karslruhe Institute of Technology (KIT, Alemania) y Wei Zhan y Masayoshi Tomizuka en la University of California, Berkeley (UCB, Estados Unidos). Se suele decir que unas veces se gana y otras se aprende, y yo en estas estancias quizás aprendí demasiado ... No obstante, me guardo grandísimos momentos (admirar las secuoyas gigantes o hacer mi primera escalada en roca entre ellos) y amigos, como Su Shaoshu o Frank Bieder, con los que aún guardo un cierto contacto.

A mis compañeros, mejor dicho, amigos, de laboratorio: Javier Araluce, Rodrigo, Felipe (chavalín), Santiago, Miguel Antunes, Miguel Eduardo, antiguos compañeros como Javier del Egido, Óscar, Alejandro, Eduardo, Roberto y Pablo Alcantarilla, y a nuevos becarios como Fabio, Navil y Pablo. Gracias de corazón por estar ahí, en nuestras charlas sobre tecnología, empleos, el camino correcto a seguir y la vida en general.

Especial mención vuelvo a hacer a Miguel Eduardo y mi compañero Marcos Conde, cuya apoyo intenso ayudó a centrar mi camino en técnica y escritura. Muchísimas gracias por todo lo que aprendí a vuestro lado. Especial mención a mi querido profesor Ángel Álvarez, por sus valiosos consejos para afrontar mi carrera profesional y mi vida en general de la mejor forma posible. Eres muy grande.

A mis amigos de la universidad, especialmente a Rocío, Juan Carlos, Esther, Sergio, Pablo, Rubén y Adrián Rocandio. Aún me acuerdo de cuando empezamos con la carrera y como el paso inexorable del tiempo nos moldea a conveniencia. Os deseo lo mejor en vuestro futuro.

A mis buenos amigos Samuel y Adrián, con quien gran parte de mi vida he compartido. Con especial cariño guardo las interminables charlas sobre la vida y el futuro después de Karate, de comer, de cenar, en el coche, siempre quejándonos de la hora que marcaba el reloj al final de tan interminables conversaciones.

A mi familia, uno de los pilares de mi vida. A mi padre Juan Antonio y a mi madre Petra, que en paz descanse, les debo todo lo que soy y es por ello por lo que les estaré siempre agradecido. Querido padre, gracias por ser tan Genaro, arisco y pesado. Siempre has sido mi ejemplo a seguir, aunque cuando tenga 42 años me sigas regañando por subir con las zapatillas puestas. Querida madre, no se muere quien se va, sólo se muere el que se olvida, y tú nunca caerás en el olvido. A mi querida hermana-calili-chessmaster Silvia, con quien tantas regañinas he tenido, pero el cariño que nos tenemos las supera a todas. A mi perrita Nuka (a.k.a. Dragón, ChupaChups cuando le cortamos el pelo o Nuki-Nuki), cuyos paseos matutinos son probablemente el ingrediente secreto para la elaboración de esta tesis, dando rienda suelta a mi cabeza para imaginar nuevas propuestas mientras miraba el cielo azul. A mi querido *experimental setup* que me ha acompañado durante toda mi vida académica: silla de madera de la cocina, ratón de Hello Kitty tomado prestado de mi hermana y mi querido portátil táctil. Sois la base de todo mi trabajo.

Al resto de la familia, amigos, compañeros, entrenadores y profesores, gracias por todo.

Y, por último, la persona más importante de mi vida ahora mismo. Mi querida Marta, la persona más maravillosa y buena que conozco. Hemos compartido risas, lloros, besos y abrazos. Nunca me cansaré de repetirte lo suave que tienes la piel tras darte un beso en la mejilla y después hacerte de rabiar. Espero que esta situación esté dentro de un while cuya condición sea *True*.

"Te quiero más que ayer, pero menos que mañana. Hoy, y siempre"

Mi querido lector, disculpa mi monólogo de agradecimientos, es mi forma de ser y la cual tengo por bandera, aunque creo que ha quedado bonito. Podría decir mil anécdotas más de mi doctorado, pero como diría Aragorn, legítimo Rey de Gondor, enfrente de la mismísima Puerta Negra: *Hoy no es ese día*.

Vamos a la lectura importante, que empiece el *Rock and Roll* !!

Resumen

TBD

Palabras clave: Autonomous Driving, Deep Learning, Motion Prediction, Scene Understanding.

Abstract

TBD

Keywords: Autonomous Driving, Deep Learning, Motion Prediction, Scene Understanding.

Contents

List of Figures

List of Tables

List of Source Code

List of Acronyms

AD	Autonomous Driving.
ADS	Autonomous Driving Stack.
AI	Artificial Intelligence.
BEV	Bird's Eye View.
CMP	Conditional Motion Prediction.
DL	Deep Learning.
ITS	Intelligent Transportation Systems.
MP	Motion Prediction.
PMP	Passive Motion Prediction.
SOTA	State-of-the-Art.

Chapter 1

Introduction

Aaay, el oro, la fama, el poder.
Todo lo tuvo el hombre que en su día se autoproclamó
el rey de los piratas, ¡GOLD ROGER!
Mas sus últimas palabras no fueron muy afortunadas:
"¿¡MI TESORO!? Lo dejé todo allí, buscadlo si queréis,
ojalá se le atragante al rufián que lo encuentre.

Opening 1 de One Piece: "We are"
Autor original: Hiroshi Kitadani

1.1. Motivation

Autonomous Driving (AD) have held the attention of technology enthusiasts and futurists for some time as evidenced by the continuous research and development in Intelligent Transportation Systems (ITS) over the past decades, being one of the emerging technologies of the *Fourth Industrial Revolution*, and particularly of the Industry 4.0.

The concept *Fourth Industrial Revolution* or Industry 4.0 was first introduced by Klaus Schwab , CEO (Chief Executive Officer) of the World Economic Forum, in a 2015 article in Foreign Affairs (American magazine of international relations and United States foreign policy). A technological revolution can be defined as a period in which one or more technologies are replaced by other kinds of technologies in a short amount of time. Hence, it is an era of accelerated technological progress featured by Researching, Development and Innovation whose rapid application and diffusion cause an abrupt change in society. In particular, the *Fourth Industrial Revolution* conceptualizes rapid change to industries, technology, processes and societal patterns in the 21st century due to increasing inter-connectivity and smart automation. This industrial revolution focuses on operational efficiency, being the following four themes which summarize it:

- Decentralized decisions: Ability of cyber physical systems to make decisions on their own and to perform their tasks as autonomously as possible.
- Information transparency: Provide operators with comprehensive information to make decisions. Inter-connectivity allows operators to gather large amounts of information and data from all points in the manufacturing process in order to identify key areas or aspects that can benefit from improvement to enhance functionality.
- Technical assistance: Ability to assist humans with unsafe or difficult tasks and technological facility of systems to help humans in problem-solving and decision-making.
- Interconnection: Ability of machines, sensors, devices and people to communicate and conect with each other via the Internet of Things (IoT) or the Internet of People (IoP).

Based on the aforementioned principles, this revolution is expected to be marked by breakthroughs in emerging technologies in fields such as nanotechnology, quantum computing, 3D printing, Internet of Things (IoT), fifth-generation wireless technologies (5G), Robotics, Computer Vision (CV), Artificial Intelligence (AI) or the scope of this PhD thesis, Autonomous Driving Stacks (ADSs). The sum of all these advances are resulting in machines that can potentially see, hear and what is more important, think, moving more deftly than humans.

An ADS, also referred in the literature as Intelligent Vehicle (IV), driverless car or autonomous car, is a vehicle tan can sense its surrounding and moving safely with little or even no human input. These ADSs must combine a variety of sensors to understand the traffic scenario, like RADAR (RAdio Detection A Ranging), LiDAR (Light Detection and Ranging), cameras, Inertial Measurement Unit (IMU), wheel odometry, GNSS (Global Navigation Satellite System) or ultrasonic sensors, and detect, track and predict (which is the main purpose of this thesis) the most relevant obstacles around the ego-vehicle. Then, advanced control and planning systems process this sensory information in combination with a predefined global route to calculate the corresponding control commands to drive throughout the environment, ensuring a safe driving.

The dream of seeing fleets of ADSs efficiently delivering goods and people to their destination has fueled billions of dollars and captured consumer's imaginations in investment in recent years. Nevertheless, according to the "Autonomous driving's future: Convenient and connected" report, published by the global management consulting firm McKinsey & Company in January 2023, even after some setbacks have pushed out timelines for AD launches and delayed customer adoption, the transportation community still broadly agrees that AD has the potential to transform consumer behaviour, transportation and society at large. AD is considered as one of the solutions to the before mentioned problems and one of the greatest challenges of the automotive industry today.

Statistics show that 69 % of the population in the European Union (EU), including associated states, lives in urban areas. According to the World Health Organization, nearly one third of the world population will live in cities by 2030, leading to an overpopulation in most of them. Aware of this problem, the Transport White Paper published by the European Commission in 2011 indicated that new forms of mobility ought to be proposed so as to provide sustainable solutions for people and goods safely. For example, regarding safety, it sets the ambitious goal of halving the overall number of road deaths in the EU between 2010 and 2020. Nevertheless, this goal does not seem to be easy since only in 2014 more than 25,700 people died on the roads in the EU, many of them caused by an improper behaviour of the driver on the road. A similar study made by the National Highway Traffic Safety Administration (NHTSA, tranportation organization of the United States) reported in 2015 that around 94 % of traffic accidents happen because of human error. In that sense, the existence of reliable and economically affordable ADSs are expected to create a huge impact on society affecting social, demographic, environmental and economic aspects. It can produce substantial value for the auto industry, drivers and society, making driving safer, more convenient and more enjoyable. While the human driver or not could select whether to drive, in autonomous mode hours on the road previously spent driving could be used to work, watch a funny movie or even to video call a friend. For employees with long commutes, AD might shorten the workday, increasing worker productivity. Since workers, specially those related to digital jobs or related fields, may perform their jobs from an ADS, they could more easily move further away from the office, which, in turn, could attract more people to suburbs and rural areas. Besides this, it is estimated to cause a reduction in road deaths, reduce fuel consumption and harmful emission associated and improve traffic flow, as well as an improvement in the overall driver comfort and mobility in groups with impaired faculties, such as disable or elderly people, providing them with mobility options that go beyond car-sharing services or public transportation. Other industrial applications of autonomous vehicles are agriculture, retail, manufacturing, commercial and freight transport or mining.

1.2. Historical Context

ADSs have become a challenge for auto competitions and technology companies, which has derived in an intense competition. Though today companies such as Mercedes, Ford or Tesla are racing to build ADSs for a radically changing consumer world, the research and development of autonomous robots is not new.

In 1500, centuries before the invention of the automobile, Leonardo da Vinci designed a cart that could move without being pulled or pushed. In 1868, Robert Whitehead invented a torpedo that could propel itself underwater in order to be a game-changer for naval fleets all over the world. In terms of robotic solutions for intelligent mobility, the study was started in the 1920s, being the concept of Autonomous Car defined in Futurama, an exhibit at the 1939 New York Wolrd's Fair. General Motors created the exhibit to display its vision of what the world would look like in 20 years, including an automated highway system that would guide ADS. By 1958, General Motors made this concept a reality (at least as a proof of concept) being the car's front end embedded with sensors to detect the current flowing through a wire embedded in the road. The first semi-automated car was developed in 1977 by Japan's Tsukuba Mechanical Engineering Laboratory. The vehicle reached speeds up to 30 km/h with the support of an elevated rail.

Figure 1.1: Stanley, 2005 DARPA Grand Challenge winner
Source: *Stanford university*

Nevertheless, the first truly autonomous cars appeared in the 1980s with Carnegie Mellon University's Navlab and ALV projects funded by the USA company DARPA (Defense Advanced Research Projects Agency) in 1984 and EUREKA Prometheus project (1987) developed by Mercedes-Benz and Bundeswehr University Munich's. By 1985, the ALV project had shown self-driving speeds on two-lane roads of 31 km/h with obstacle avoidance added in 1986 and off-road driving in day and night conditions by 1987. Furthermore, from the 1960s through the second DARPA Grand Challenge in 2005 (212 km off-road course near the California-Nevada state line, surpassed by all but one of the 23 finalists), automated vehicle research in the United States was primarily funded by DARPA, the US Army and US Navy, yielding rapid advances in terms of speed, car control, sensor systems and driving competence in more complex conditions. This caused a boost in the development of autonomous prototypes by companies and research organizations, most of them from the United States. Figure 1.1 shows Stanley, the 2005 DARPA Gran Challenge winner, from Stanford university.

Even though self-driving cars have not yet displaced conventional cars, there can be found several examples of how it has become a hot topic for powerful companies such as Delphi Automotive Systems, Audi, BMW, Tesla, Mercedes-Benz or Waymo.

In 2005 Delphi broke the Navlab's record achievement (driving 4,584 km while remaining 98 % of the time autonomously) by piloting an Audi, improved with Delphi technology, over 5,472 km through 15 states while remaining in self-driving mode 99 % of the time. Moreover, in 2005 the USA states of Michigan, Virginia, California, Florida, Nevada and the capital, Washington D.C., allowed the testing of automated cars on public roads.

In 2017, Audi stated that its A8 car prototype would be automated at speeds up to 60 km/h by using its perception system named "Audi AI". Also, in 2017 Waymo (self-driving technology development company subsidiary of Alphabet Inc) started a limited trial of a self-driving taxi service in Phoenix, Arizona.

Figure 1.2 shows the total number of autonomous test miles and miles per disengagement in California (Dec 2019 - Nov 2020) by some of the most important AD technology development companies around the world. The concept disengagement is quite useful to assess the quality of an ADS, defined as the deactivation of the autonomous mode when a failure of the autonomous technology is detected or when a safe operation requires that the autonomous vehicle test driver disengages the autonomous mode, resulting in control being seized by the human driver.

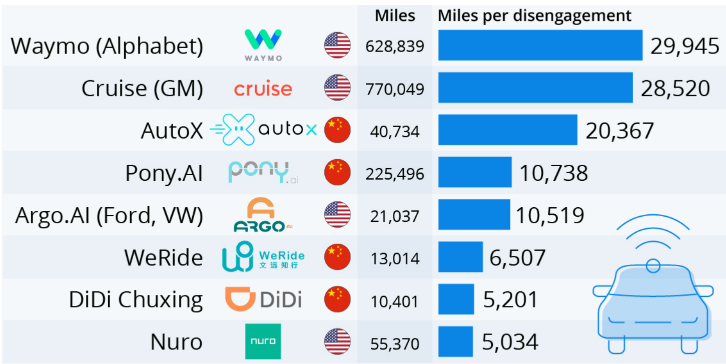

Figure 1.2: Number of autonomous test miles and miles per disengagement (Dec 2019 - Nov 2020)
Source: *DMV California, via The Last Driver License Holder*

At the moment of writing this thesis (2023), many vehicles on the road are considered to be semi-autonomous due to safety features like braking systems, assisted parking, lane boundaries detection or predict the long-term behaviour of the users around the vehicle to execute the most optimal action in a safely way. Regarding this, the Society of Automotive Engineers (SAE) published the concept of autonomy levels in 2014, as part of its "Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems" report. Figure 1.3 illustrates the six levels of autonomy (the higher the level, the more autonomous the car is), where it can be appreciated that Level Zero means "No Automation", being the acceleration, braking and steering controlled by a human driver at all times, and Level Five represents Full Automation, where there is a full-time automation of all driving tasks on any road, under any conditions, whether there is a human on board or not.

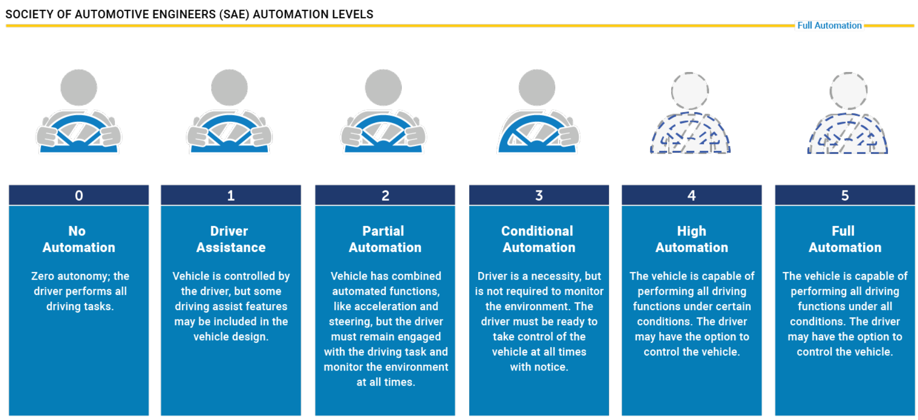

Figure 1.3: Society of Automotive Engineers (SAE) automation levels
Source: *NHTSA (National Highway Traffic Safety Administration)*

In that sense, today most vehicles only included basic Advanced Driver Assistance Systems (ADAS), but major advancements in AD capabilities are on the horizon. According to a 2021 McKinsey consumer survey, growing demand for AD systems could create billions of dollars in revenue. Based on a consumer interest in AD features and commercial solutions available on the market today, ADAS and AD could generate between \$300 and %400 billions in the passenger car market by 2035. Figure 1.4 illustrates an interesting study reporting the revenues of ADAS and AD from Level 1 (Driver Assistance) to Level 4 (High Automation). As expected, Level 5 is excluded from this study due to the huge difficulties the automotive companies would have to face to adapt their systems under totally different environmental conditions.

1.3. Autonomous Driving architecture

To sum up what commented above, increasing the level of autonomous navigation in mobile robots (from agriculture to public and private transport) are expected to create

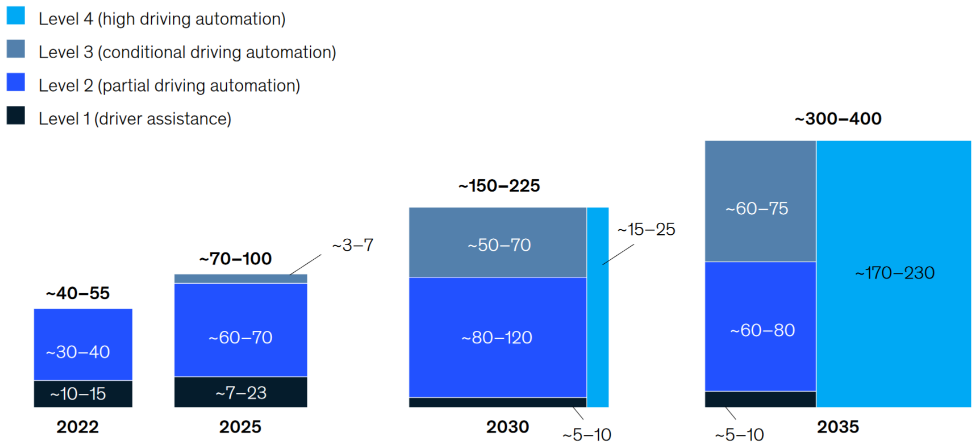

Figure 1.4: Advanced Driver Assistance systems (ADAS) and Autonomous Driving (AD) revenues in $ billion
Source: *McKinsey Center for Future Mobility*

tangible business benefits to those users and companies employing them. However, designing an autonomous navigation system does not seem to be an easy task. In the State-of-the-Art (SOTA) we can distinguish two main kind of software architectures: End-to-End and modular. Figure 1.5 illustrates the entire AD architecture starting from sensing, all the way to longitudinal (throttle/brake) and lateral (steering angle) control of the vehicle, which are the commanded signals that feed the low-level electronic system that moves the vehicle, like a drive-by-wire system [1]. End-to-End are considered black-box models, where a single neural network performs the driving task (throttle/steering/brake) from raw sensor data, in such a way the error be may vanished since intermediate representations are jointly optimized, but these are not very interpretable. On the other hand, modular architectures (considered as glass models as counterpart to End-to-End approaches) separate the driving task into individually programmed or trained modules. This solution is more interpretable, since the know-how of a research group or company is easily transferred, they allow parallel development, being the standard solution in industrial research, but the error is propagated, where intermediate representations can led to suboptimal performance. For example, incorrect object detection can lead to low-quality tracking and motion prediction.

Considering the features of the research group and main projects (Techs4AgeCar, AIVATAR) where this thesis has been developed, we integrate our algorithms in a software modular approach. An example of modular An example of modular approach is shown in Figure 1.6. Despite the fact in the literature some authors include certain modules in different layers, specially motion prediction algorithms, which are usually classified as a perception algorithm, but sometimes is included as part of the planning or decision-making layers, we can hierarchically break down (from raw data to the driving task) a standard AD architecture into the following software layers:

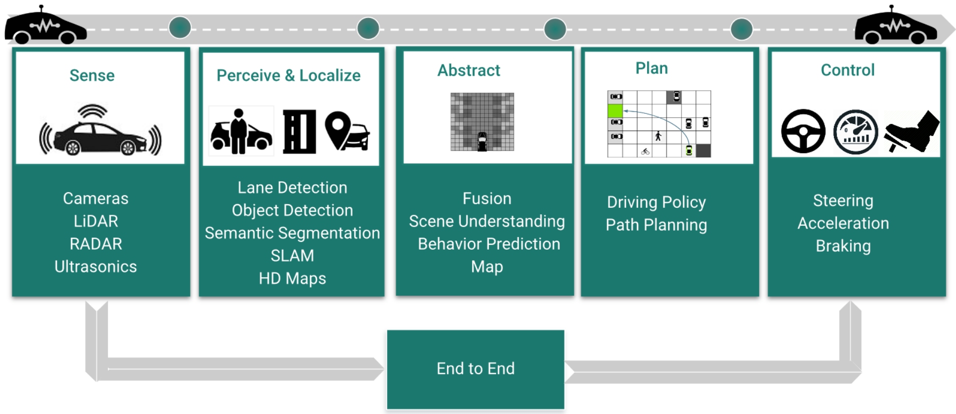

Figure 1.5: Autonomous Driving Stack (ADS) modular vs end-to-end pipeline
Source: *Vrunet: Multi-task learning model for intent prediction of vulnerable road users* [2]

- Localization layer: Positions and locates the vehicle on a map with real-time and centimetric accuracy approach. The main source of information is a robust differential-GNSS, though IMUs, wheel odometry and even cameras are commonly employed.

- Perception layer: Understand the environment around the ego-vehicle thanks to the information collected by the sensors. If defined as multi-stage, the perception layer first detects the most relevant obstacles, then track them over time to finalize long-term predict with plausible predictions. In order to perform object detection, LiDAR, camera and RADAR are the main sensors that provide the corresponding raw data. Additionally, HD map information is frequently used in the motion prediction tasks by most SOTA algorithms.

- Mapping layer: Responsible for creating a topologic, semantic and geographical modeling of the environment through which the vehicle drives, being the HD Map graph the most common source of information.

- Planning layer: This layer is comprised of three components: route, behavioural and trajectory planner. The route planner computes the most optimal (in terms of distance, time and so forth and so on) global route from some predefined start and goal. It uses the localization and mapping output. On the other hand, the behaviour planner, also referred as decision-making layer by some authors, it performs high-level decision-making of driving behaviours such as lane changes or progress through intersections, mostly focused on the previously computed global route and current localization. It can be seen as an atomization of the global route in different behaviors to reach the goal. Finally, the trajectory planner, also known as local planner, generates a time schedule for how to follow a path given constraints such as position, velocity and acceleration in order to meet the previously decided behaviour and taking into account the prediction from the perception layer, avoiding obstacles in optimal direction and speed conditions.

- Control layer: Once the local plan is calculated, the control layer is responsible for generating the commands that are sent to the actuators. It receives as input some waypoints from the calculations made in the trajectory planner. Once these waypoints are received, most authors perform spline interpolations and a velocity profile that ensures a smooth and continous trajectory.

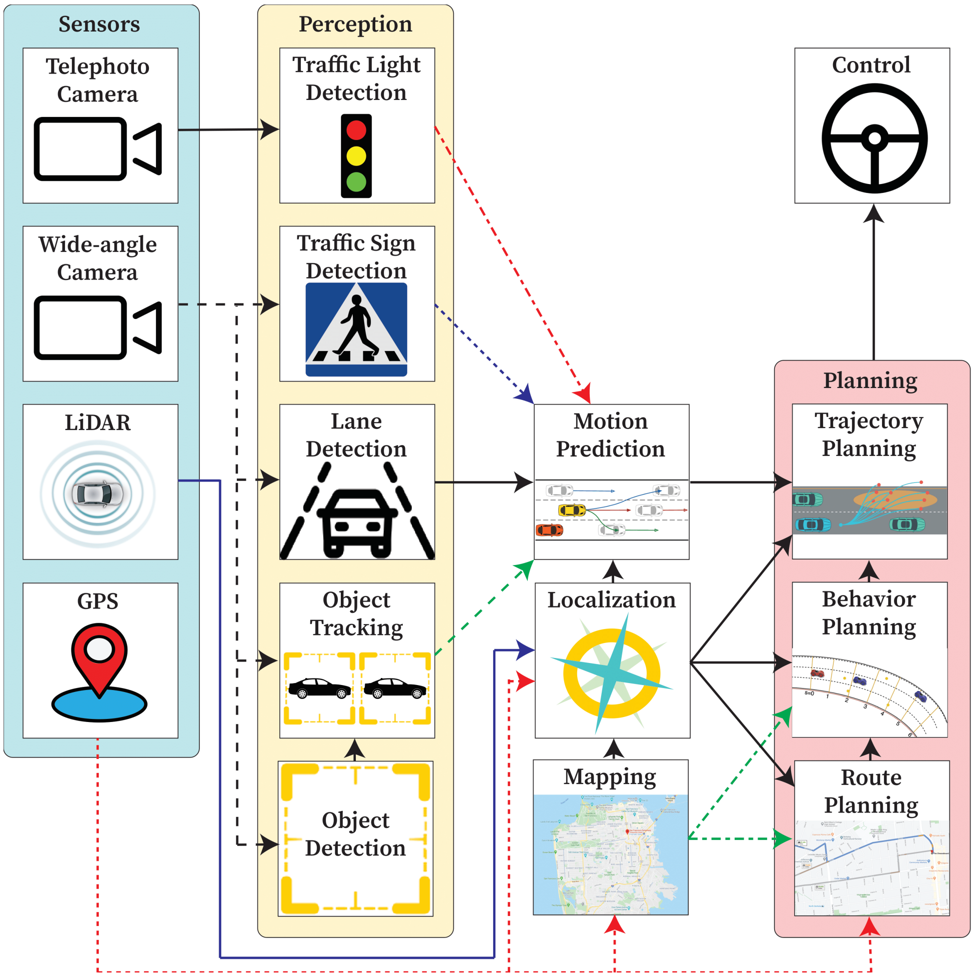

Figure 1.6: Autonomous Driving Stack (ADS) modular pipeline
Source: *Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles* [3]

1.4. Problem statement

As commented in previous sections, in order to operate efficiently and safely in highly dynamic, complex and interactive driving scenarios, ADS need to smartly reason like human beings via predicting future motions of surrounding traffic participants during navigation. Nevertheless, achieving accurate and robust Motion Prediction (MP) in one of the most difficult and interesting challenges to achieve full-autonomy, since it is equivalent to a bridge between the former stages of the perception layer, where the scene is understood detecting and tracking static and dynamic objects of the environment, and the planning a control layer, where the driving commands are sent to the vehicle. Here are some of the most important challenges:

1. Heterogeneity of traffic participants. Traffic partipants (specially those which are dynamic) can be roughly classified as cyclists, pedestrians or other vehicles. The prediction model should be capable of differentiating the motion patterns of heterogeneous traffic participants, in such a way fine-grained classification (detection module) is quite beneficial to include additional metadata along with the past observations.

2. Complexity of road structure. Road structures are highly diverse and complex, specially in highways and urban areas, which noticeably affect the motion behaviours of traffic participants.

3. Variable number of interactive agents. The prediction model must deal with a number of associated traffic participants within a certain area that can vary from time to time, such as intersections or roundabouts. Then, while driving, a comprehensive representation of the scene must be able to accommodate an arbitrary number of involved traffic participants.

4. Multimodality of driving behaviours. In real-world, despite we know the behaviour our vehicle will carry out, the motion patterns of other traffic participants can be considered inherently multimodal since there is usually more than one reasonable option for a driver to choose, specially in intersections, when the number of lanes increases or even in the same lane with different velocity profiles (constant velocity, sudden break, sudden acceleration). In that sense, a robust and reliable MP model is expected to be human-like and capture different plausible motion modalities where an agent can travel in the prediction horizon.

5. Complex interpendencies among traffic participants and road infrastructure. Agent-Agent, Agent-Road and Road-Road interpendencies are of great importance for MP and interaction modeling, even more taking into account the complexity of road structures and heterogeneity of traffic participants aforementioned. As expected, an agent future trajectory will be affected not only by its own past trajectory and driving objectives (given by the behaviour planner) but also by other surrounding agents past trajectories, traffic rules and physical constraints.

1.5. Objectives and Structure of this work

The main scope of this thesis is to study the SOTA and development of novel and efficient interaction-aware Deep Learning based MP models, focusing on long-term (from 3 to 6 s) prediction horizon and AD, where traffic participants can range from trucks to pedestrians, instead of models focused on pedestrian trajectory prediction. The main inputs that will be used throughout this work are the physical (map) information and historical states (that may include agent position, velocity, orientation, object type and category) of traffic participants in Bird's Eye View (BEV), assuming these objects have been previously tracked by our ego-vehicle (also referred as the autonomous car). Though the evaluation of these methods will be done using a single target agent, as proposed by some of the most important prediction datasets, like Argoverse 1 [4] and Argoverse 2 [5], some of the proposed methods will be trained considering multi-agent. In this thesis, the solutions to the aforementioned challenges will be discussed and investigated progressively. In order to achieve the main scope, the following objectives will be met:

1. Research of SOTA MP, focused on Deep Learning (DL) and the AD paradigm.
2. Propose of several MP architectures, studying the progressive incorporation of DL mechanisms and different sources of information and metadata, achieving SOTA accuracy while reducing in millions of parameters previous models as well as inference time.
3. Validate the proposed models in downstream applications, such as decision-making or behavioural planning, taking into account former stages of the perception layer (detection and tracking) instead of static files (benchmarks) in hyper-realistic simulation, as a preliminary stage before implementing it in a real-world vehicle.

The organization of this document has been done as follows:

- Chapter 2 reviews the most important features and methods of physics-based and learning-based MP methods. The physics-based methods are reviewed according to a taxonomy similar to existing reviews. The learning-based methods are reviewed based on two classification criteria: scene representation and trajectory decoding.
- Chapter 3 presents a technical background, mostly focused on DL mechanisms to deal with temporal sequences and interactions, to deeply understand the proposed methods.
- Chapter 4 illustrates the different prediction models developed in the thesis using different validation environments, from unimodal physic-based prediction to the final model of the thesis which takes into account agents interactions, map information and past observations using a novel scene representation with heuristic proposals, graph-based encoding, DL-based goal proposals and motion refinement.
- Chapter 5 addresses the integration of the final model of the thesis with upstream and downstream modules to contribute the entire pipeline and closed-loop for AD.
- Chapter 6 summarizes the thesis and provides some promising directions for future work in the areas of MP and validation.

Chapter 2

Related Works

Llegaré a ser el mejor, El mejor que habrá jamás
Mi causa es ser su entrenador, Tras poderlos capturar.
Viajaré a cualquier lugar, Llegaré a cualquier rincón
Y al fin podré desentrañar, El poder de su interior.
¡Pokémon! Hazte con todos (solos tú y yo),
Es mi destino, mi misión
¡Pokémon! Tú eres mi amigo fiel,
Nos debemos defender.

Opening 1 de Pokémon: "Gotta catch 'em all!"
Autor original: Jason Paige

2.1. Introduction

One of the crucial tasks that ADSs must face during navigation, specially in arbitrarily complex urban scenarios, is to predict the behaviour of dynamic obstacles [4], [6]. In a similar way to humans that pay more attention to nearby obstacles and upcoming turns than considering the obstacles far away, the perception layer of an ADS must focus more on the salient regions of the scene, particularly on the more relevant dynamic agents to predict their future behaviour before conducting a maneuver, such as lane changing or accelerating. In that sense, before proceeding with the study of the different methods of the SOTA of MP in the field of AD, one important thing to note is that this thesis is focused on non-conditional motion prediction, also referred as Passive Motion Prediction (PMP), where the prediction of surrounding agents is not influenced by the future decisions of the ego-vehicle or even other agents, referred as Conditional Motion Prediction (CMP) in the literature. Most existing works [7]–[12] focus on a passive prediction scheme, where the future states of a particular agent are predicted given its past information, other surrounding agents information and interactions as well as the physical context. Then, downstream planning modules, specially the behaviour planning module (also referred as decision-making layer, as stated in Section 1.3), the ego-vehicle (our vehicle) future actions are computed according to the predicted trajectories in a passive manner, that is, without modifying the output of the prediction model, and the global route previously calculated.

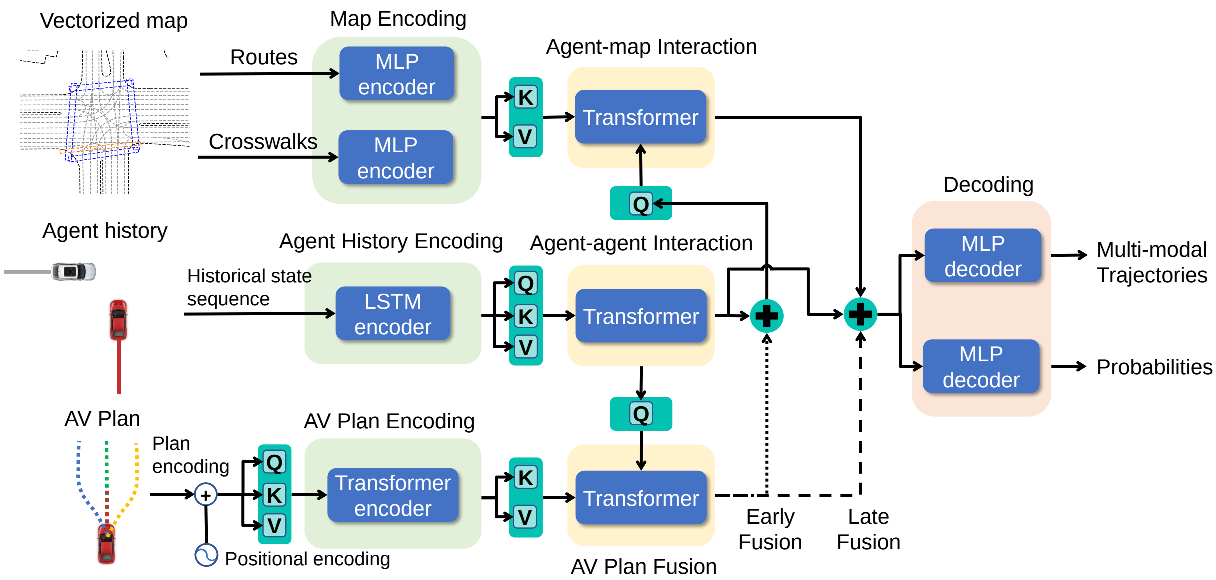

Figure 2.1: Example of a Conditional Motion Prediction (CMP) network.
We highlight the influence of the AD in the prediction of surrounding agents.
Source: *Conditional Predictive Behavior Planning with Inverse Reinforcement Learning for Human-like Autonomous Driving* [13]

Nevertheless, to ensure safety under various predicted trajectories of the surrounding agents, our ADS must overly conservative with inefficient maneuvers, specifically in arbitrarily complex traffic scenarios, because PMP models ignore the fact that the future states of an agent can influence the future actions of other agents, what is the most realistic situation. To this end, researchers recently started to explore a more coherent interactive prediction and planning framework which relies on predicting the surrounding agents future trajectories conditioned on the ego-vehicle future actions [14] [15] [16], as a preliminary state to implement a fully-interaction graph where the future states of all agents (either autonomous prototypes or human-driven) influence in the decision of all agents. Under such frameworks, the ADS can reason over potential actions while considering its influence on surrounding agents, as observed in Figure 2.1, inducing less conservative and more efficient maneuvers in highly interactive scenarios. [13] propose a learning-based behaviour planning framework that learns to predict conditional multi-agent future trajectories, evaluating decisions from real-world human data. Moreover, they propose a two-stage learning process where the prediction model is trained first conditioned on the ADS future actions, and then used as an environment model in the learning of the cost function with maximum entropy Inverse Reinforcement Learning (IRL). [17] argue that CMP-based models essentially learns the posterior distribution of future trajectories conditioned on the future states of the ego-vehicle, where this future trajectory is treated as an observation, whilst safe and realistic prediction models should build the MP to approximate the future trajectory distribution under the intervention of enforcing the ADS future states, referring this new task as Interventional Behaviour Prediction (IBP). As aforementioned, the algorithms studied and developed throughout this thesis do not focus on the joint study of the prediction and behaviour planning modules, but on building efficient and powerful PMP algorithms without considering the future states of the autonomous agents as an additional condition.

Once the differences between CMP and PMP have been illustrated, we proceed with the problem formulation, main contextual factors and classification of prediction methods.

2.2. Problem Formulation of Motion Prediction

Given a sequence of past trajectories $a_P = [a_{-obs'_{len}+1}, a_{-obs'_{len}+2}, ..., a_0]$ for an agent, we aim to predict its future steps $a_F = [a_1, a_2, ..., a_{pred_{len}}]$ up to a fixed time step $pred_{len}$. Running in a specific traffic scenario, each agent will interact with static HD maps m and the other dynamic actors, meeting the corresponding traffic and social rules. Therefore, the probabilistic distribution that we want to capture is $p(a_F|m, a_P, a_P^O)$, where a_P^O denotes the other agents observed states. The output of most existing methods is a set of trajectories $A_F = \{a_F^k\}_{k\in[0,K-1]} = \{(a_1^k, a_2^k, ..., a_{pred_{len}}^k)\}_{k\in[0,K-1]}$ for each agent, where K represents the number of modes or plausible future directions, due to the inherent uncertainty associated to the prediction problem. This set of trajectories for each agent will be used by downstream decision modules. On top of that, TNT (Target-driven trajectory prediction) [18] is one of the first methods that introduces specific preliminary future positions in the problem formulation, also referred as goals, being TNT [18]-like methods distribution approximated as:

$$\sum_{\tau\in T(m,a_P,a_P^O)} p(\tau|m, a_P, a_P^O)p(a_F|\tau, m, a_P, a_P^O) \tag{2.1}$$

where $T(m, a_P, a_P^O)$ is the space of candidate goals depending on the driving context.

However, the map space m is large, and the goal space $T(m, a_P, a_P^O)$ requires careful design. In that sense, some methods expect to accurately predict the actor motion by extracting good features. For example, LaneGCN [12] tries to approximate $p(a_F|m, a_P, a_P^O)$ by modeling $p(a_F|M_{a_0}, a_P, a_P^O)$, where M_{a_0} is a "local" map features that is related to the actor's state a_0 at final observed step $t = 0$. To extract M_{a_0}, they use a_0 as an anchor to retrieve its surrounding map elements and aggregate their features. We found that not only the "local" map information is important, but also the goal area maps information is of great importance for accurate trajectory prediction. So, we reconstructed the probability as:

$$\sum_{\tau} p(\tau|M_{a_0}, a_P, a_P^O)p(M_\tau|m, \tau)p(a_F|M_\tau, M_{a_0}, a_P, a_P^O) \tag{2.2}$$

We directly predict possible goals τ based on actors' motion histories and driving context. Therefore, GANet is genuinely end-to-end, adaptive, and efficient. Then, we apply the predicted goals as anchors to retrieve the map elements in goal areas explicitly and aggregate their map features as M_τ.

2.3. Contextual Factors and Outputs of Motion Prediction

2.4. Classification of Motion Prediction methods

Most traditional predictions methods [19], which usually only consider physics-related factors (like the velocity and acceleration of the target vehicle that is going to be predicted) and road-related factors (prediction as close as possible to the road centerline), are only suitable for **short-time** prediction tasks [19] and simple traffic scenarios, such as constant velocity (CV) in a highway or a curve (Constant Turn Rate Velocity, CTRV) where a single path is allowed, i.e. multiple choices computation are not required. Recently, MP methods based on DL have become increasingly popular since they are able not only to take into account these above-mentioned factors but also consider interaction-related factors (like agent-agent [20], agent-map [21] and map-map [12]) in such a way the algorithm can adapt to more complex traffic scenarios (intersections, sudden breaks and accelerations, etc.). It must be consider that multimodal, specially in the field of vehicle motion prediction, does not refer necessarily to different directions (e.g. turn to the left, turn to the right, continue forward in an intersection), but it may refer to different predictions in the same direction that model a sudden positive or negative acceleration, so as to imitate a realistic human behaviour in complex situations. As expected, neither classical nor Machine Learning (ML) methods can model these situations [19].

2.4.1. Physic-based Motion Prediction

2.4.2. Machine Learning based Motion Prediction

2.4.3. Deep Learning based Motion Prediction

2.4.4. Reinforcement Learning based Motion Prediction

Table 2.1: Main state-of-the-art methods for Motion Prediction. Main categories are Encoder (splitted into motion history, social info (agent interactions) and map info (physical information)), Decoder, Output representation and Distribution over future trajectories.

Method	Encoder			Decoder	Output	Trajectory Distribution
	Motion history	Social info	Map info			
SocialLSTM [22]	LSTM	spatial pooling	–	LSTM	states	samples
SocialGan [20]	LSTM	maxpool	–	LSTM	states	samples
Jean [23]	LSTM	attention	–	LSTM	states	GMM
TNT [18]	polyline	maxpool, attention	polyline	MLP	states	weighted set
LaneGCN [12]	1D-conv	GNN	GNN	MLP	states	weighted set
WIMP [16]	LSTM	GNN+attention	polyline	LSTM	states	GMM
VectorNet [24]	polyline	maxpool, attention	polyline	MLP	states	unimodal
SceneTransformer [25]	attention	attention	polyline	attention	states	weighted set
HOME [7]	raster	attention	raster	conv	states	heatmap
GOHOME [8]	1D-conv+GRU	GNN	GNN	MLP	states	heatmap
MP3 [26]	raster	conv	raster	conv	cost function	weighted samples
CoverNet [27]	raster	conv	raster	lookup	states	GMM w/ dyn. anch.
DESIRE [28]	GRU	spatial pooling	raster	GRU	states	samples
MFP [14]	GRU	RNNs+attention	raster	GRU	states	samples
MANTRA [29]	GRU	–	raster	GRU	states	samples
PRANK [30]	raster	conv	raster	lookup	states	weighted set
IntentNet [21]	raster	conv	raster	conv	states	unimodal
Multimodal [31]	raster	conv	raster	conv	states	weighted set
MultiPath [32]	raster	conv	raster	MLP	states	GMM w/ static anchors
MultiPath++ [9]	LSTM	RNNs+maxpool	polyline	MLP	control poly	GMM
PLOP [33]	LSTM	conv	raster	MLP	state poly	GMM
Trajectron++[6]	LSTM	RNNs+attention	raster	GRU	controls	GMM
CRAT-PRED[11]	LSTM	GNN+attention	–	MLP	states	weighted set
R2P2 [34]	GRU	–	polyline	GRU	motion	samples
DKM [35]	raster	conv	raster	conv	controls	weighted set

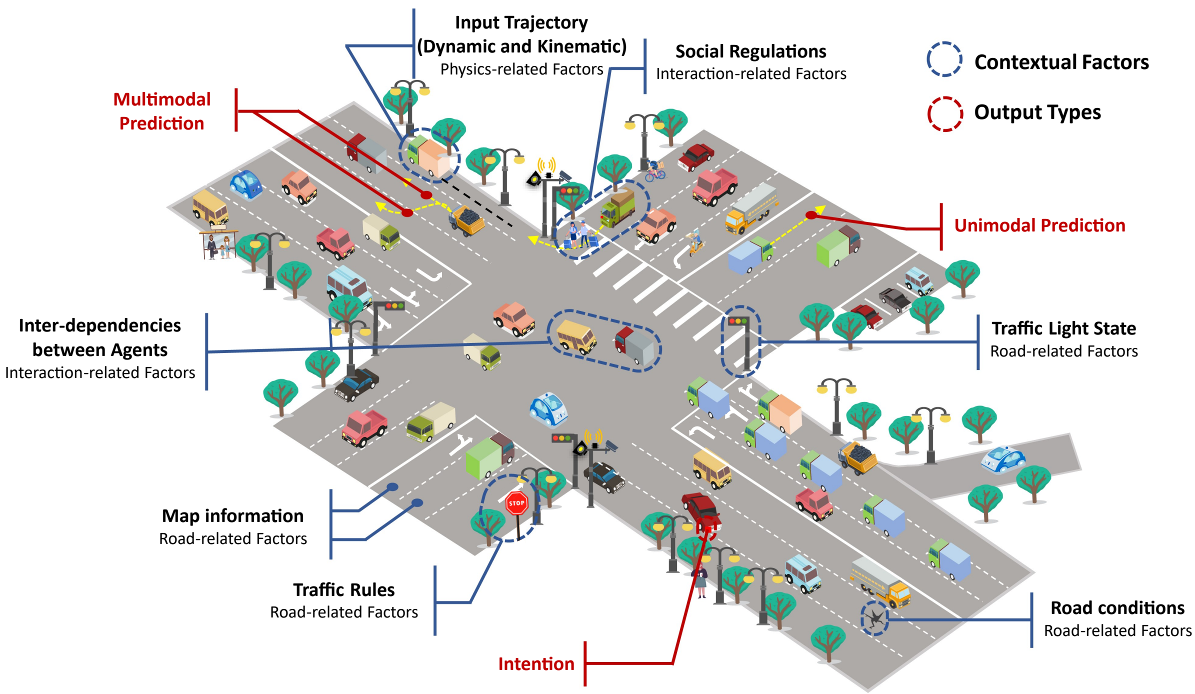

Figure 2.2: Contextual factors and output types in Vehicle Motion Prediction
Source: *A survey on trajectory-prediction methods for autonomous driving* [19]

Chapter 3

Theoretical Background

Desde que el mundo cambió, estamos mucho más unidos
con los Digimon, luchamos juntos contra el mal.
Algo extraño pasaba, Digievolucionaban,
en tamaño y color, ellos son los Digimon.

Opening 1 de Digimon: "Butterfly"
Autor original: Kōji Wada

3.1. Kalman Filtering

https://arxiv.org/pdf/1710.04055.pdf

3.2. Physic-based Motion models

The estimation of a vehicle's dynamic state is one of the most fundamental data fusion tasks for intelligent traffic applications. For that, motion models are applied in order to increase the accuracy and robustness of the estimation. This paper surveys numerous (especially curvilinear) models and compares their performance using a tracking tasks which includes the fusion of GPS and odometry data with an Unscented Kalman Filter. For evaluation purposes, a highly accurate reference trajectory has been recorded using an RTK-supported DGPS receiver. With this ground truth data, the performance of the models is evaluated in different scenarios and driving situations.

Vehicle tracking is one of the most important data fusion tasks for Intelligent Transportation Systems (ITS). Especially for advanced driver assistance systems such as Collision Avoidance/Collision Mitigation (CA/CM), Adaptive Cruise Control (ACC), Stop-and-Go-Assistant, or Blind Spot Detection, a reliable estimation of other vehicles' positions is one of the most critical requirements.

In order to increase the stability and accuracy of the estimation, the vehicles are mostly assumed to comply with certain motion models which describe their dynamic behavior.

Another advantage of this approach is the ability to predict the vehicle's position in the future (which can for instance be used to calculate a collision probability). From the data fusion point of view, the task is to estimate the parameters of the model - taking into account all available observations. The most common approach for this task is the Kalman Filter or one of its derivates [1].

The application of motion models has been intensively studied for a variety of ITS applications, for instance radar tracking [2] or navigation [3]. However, even applications which are from a superficial point of view not concerned by vehicle tracking often require a reliable estimation of the ego vehicle's motion in order to compensate estimates of tracked objects accordingly (an example which illustrates this is motion based pedestrian recognition [4]). Thus, the term vehicle tracking in this paper refers to the task of estimating the model parameters of either the ego vehicle or vehicles in its surrounding.

In the past, numerous motion models (with different degrees of complexity) have been proposed for this task. Some authors also compared different motion models for a certain applications in a rather general way using simulated data (e. g. [5]). However, the question which motion model is most suitable for describing vehicles' motions in certain scenarios has not yet been sufficiently answered and will therefore be the subject of this paper. In particular, an evaluation approach is proposed which is based on the combination of GPS and odometry measurements. By comparing the estimates of every model with a highly accurate reference trajectory, the filters' performances can be compared and evaluated.

The paper is organized as follows: Section II surveys the most common motion models and their state transition equations. In the following section, the methodology for evaluating the models is described. Finally, the results of the comparison are presented and discussed in section IV.

As indicated above, the models proposed in literature are numerous. A first systematization can be achieved by defining different levels of complexity. At the lower end of such a scale, linear motion models are situated. These models assume a constant velocity (CV) or a constant acceleration (CA). Their major advantage is the linearity of the state transition equation which allows an optimal propagation of the state probability distribution. [1] On the other hand, these models assume straight motions and are thus not able to take rotations (especially the yaw rate) into account.

A second level of complexity can be defined by taking rotations around the z-axis into account. The resulting models are sometimes referred to as curvilinear models. They can be further divided by the state variables which are assumed to be constant. The most simple model of this level is the Constant Turn Rate and Velocity (CTRV) model, which is commonly used for airborne tracking systems [6].[2] By defining the derivative of the velocity as the constant variable, the Constant Turn Rate and Acceleration (CTRA) model can be derived. Both CTRV and CTRA assume that there is no

[1] However, note that the measurement equation is necessarily nonlinear if the orientation angle is included in the state vector.

[2] Note that in literature, this model is sometimes referred to as CTR However, in order to obtain a consistent nomenclature, CTRV will be consequently used throughout this paper.

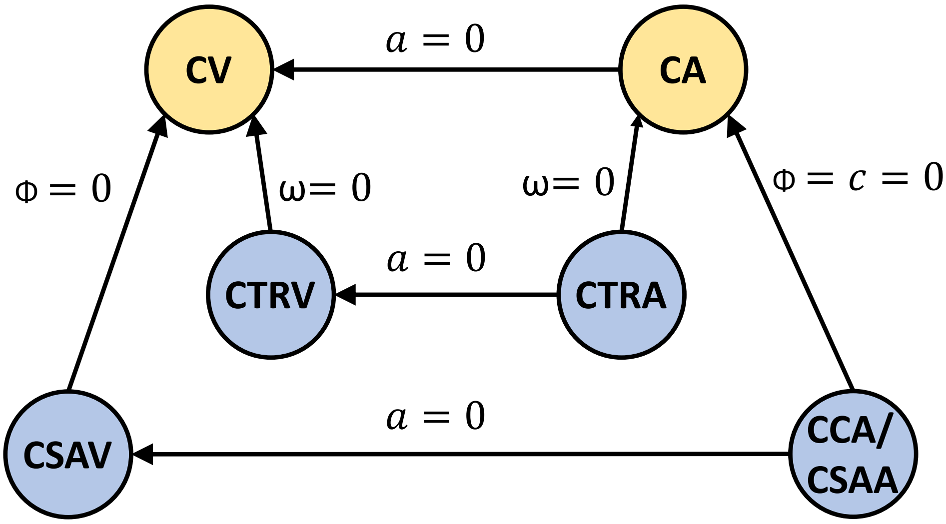

Figure 3.1: Overview about linear and curvilinear motion models. Every sophisticated model can be transformed into a simpler one by setting one state veriable to zero
Source: *Comparison and evaluation of advanced motion models for vehicle tracking* [36]

Fig. 1. Overview about linear and curvilinear motion models. Every sophisticated model can be transformed into a simpler one by setting one state veriable to zero.

correlation between the velocity v and the yaw rate ω. As a consequence, disturbed yaw rate measurements can change the yaw angle of the vehicle even if it is not moving.

In order to avoid this problem, the correlation between v and ω can be modeled by using the steering angle Φ[3] as constant variable and derive the yaw rate from v and Φ. The resulting model is called Constant Steering Angle and Velocity (CSAV). Again, the velocity can be assumed to change linearly, which leads to the Constant Curvature and Acceleration (CCA) model. [4] The connections between all models described so far are illustrated in figure 1.

From a geometrical point of view, nearly all curvilinear models are assuming that the vehicle is moving on a circular trajectory (either with a constant velocity or acceleration). The only exception is the CTRA model which models a linear variation of the curvature and thus assumes that the vehicle is following a clothoid.

While in theory curvilinear models describe the motion of road vehicles very accurately, errors may result from highly dynamic effects such as drifting or skidding. While models which are able to cope with such effects do exist (e. g. [7]), they will not be considered here for two reasons: Firstly, most ITS applications are designed for scenarios with non-critical dynamics. Furthermore, the information which are necessary for estimating the additional parameters (e. g. slip from every tire, lateral acceleration) are not observable by exteroceptive sensors. Thus, such models can be used for estimating the ego vehicle's motion, only.

3.3. B. State Transition Equations

Many of the described models (with the exception of CCA) are well-known and will thus be treated very briefly. Further details can be found in [1].

[3] This angle is defined between the axis of motion and the direction of the front wheels.

[4] If the steering angle would be used as a state variable instead of the curvature, the model could also be named Constant Steering Angle and Velocity (CSSA). From an algorithmic point of view, however, both names refer to the same model. 1) CV : As the CV model with the state space

$$\vec{x}(t) = \begin{pmatrix} x & v_x & y & v_y \end{pmatrix}^T$$

is a linear motion model, the linear state transition

$$\vec{x}(t+T) = A(t+T)\vec{x}(t)$$

is substituted by the state transition function vector

$$\vec{x}(t+T) = \begin{pmatrix} x(t) + Tv_x \\ v_x \\ y(t) + Tv_y \\ v_y \end{pmatrix}$$

in order to use it within the Unscented Kalman Filter framework.

2. CTRV: The state space

$$\vec{x}(t) = \begin{pmatrix} x & y & \theta & v & w \end{pmatrix}^T$$

can be transformed by the non-linear state transition

$$\vec{x}(t+T) = \begin{pmatrix} \frac{v}{\omega}\sin(\omega T + \theta) - \frac{v}{\omega}\sin(\theta) + x(t) \\ -\frac{v}{\omega}\cos(\omega T + \theta) + \frac{v}{\omega}\sin(\theta) + y(t) \\ \omega T + \theta \\ v \\ \omega \end{pmatrix}.$$

3. CTRA: The state space of this models expands the last one by a :

$$\vec{x}(t) = \begin{pmatrix} x & y & \theta & v & a & w \end{pmatrix}^T .$$

The state transition equation for this model is:

$$\vec{x}(t+T) = \begin{pmatrix} x(t+T) \\ y(t+T) \\ \theta(t+T) \\ v(t+T) \\ a \\ \omega \end{pmatrix} = \vec{x}(t) + \begin{pmatrix} \Delta x(T) \\ \Delta y(T) \\ \omega T \\ aT \\ 0 \\ 0 \end{pmatrix}$$

with

$$\begin{aligned} \Delta x(T) =& \frac{1}{\omega^2}[(v(t)\omega + a\omega T)\sin(\theta(t) + \omega T) \\ &+ a\cos(\theta(t) + \omega T) \\ &- v(t)\omega \sin\theta(t) - a\cos\theta(t)] \end{aligned}$$

and

$$\begin{aligned} \Delta y(T) =& \frac{1}{\omega^2}[(-v(t)\omega - a\omega T)\cos(\theta(t) + \omega T) \\ &+ a\sin(\theta(t) + \omega T) \\ &+ v(t)\omega \cos\theta(t) - a\sin\theta(t)] \end{aligned}$$

4. CCA: The state space

$$\vec{x}(t) = \begin{pmatrix} x & y & \theta & v & a & c \end{pmatrix}^T$$

is similar the one of the CTRA model, except that the yaw rate ω is replaced by the curvature $c = R^{-1}$, where R represents the radius the vehicle is currently driving. Because of

$$R = \frac{1}{c} = -\frac{v(t)}{\omega(t)} = \text{ const. } .$$

and

$$v(t) = v\left(t_0\right) - at$$

the yaw rate becomes a function of time

$$\omega(t) = (-v(t_0) - at)\, c$$

The continuous system can be described by

$$\overrightarrow{\dot{x}}(t) = \begin{pmatrix} v(t)\cos(\omega(t)t + \theta(t_0)) \\ v(t)\sin(\omega(t)t + \theta(t_0)) \\ \omega(t)t \\ a \\ 0 \\ 0 \end{pmatrix}$$

Using the equations 12 and 13 , the final system follows to

$$\overrightarrow{\dot{x}}(t) = \begin{pmatrix} (v_0 + at)\cos((-v_0 - at)\, ct + \theta_0) \\ (v_0 + at)\sin((-v_0 - at)\, ct + \theta_0) \\ (-v_0 - at)\, c \\ a \\ 0 \\ 0 \end{pmatrix}$$

The discrete state transition equation arises from integrating the continuous one

$$\vec{x}(t+T) = \int_t^{t+T} \overrightarrow{\dot{x}}(t)dt + \vec{x}(t)$$

which leads to the state transition equation 17 with

$$\begin{aligned} \gamma_1 &= \frac{1}{4a}\left(cv^2 + 4a\theta\right) \\ \gamma_2 &= cTv + cT^2a - \theta \\ \eta &= \sqrt{2\pi}vc \\ \zeta_1 &= (2aT + v)\sqrt{\frac{c}{2a\pi}} \\ \zeta_2 &= v\sqrt{\frac{c}{2a\pi}}, \\ C(\zeta) &= \int_0^{\zeta} \cos\left(\frac{\pi}{2}x^2\right) dx \end{aligned}$$

and

$$S(\zeta) = \int_0^{\zeta} \sin\left(\frac{\pi}{2}x^2\right) dx$$

Since equations 23 and 24 represent the fresnel integrals [8], a numerical approximation is used for calculating their values.

3.4. Convolutional Neural Networks

3.5. Recurrent Neural Networks

3.6. Generative Adversarial Networks

It then applies a function to generate $\mathbf{x}' = G(\mathbf{z})$. The goal of the generator is to fool the discriminator to classify $\mathbf{x}' = G(\mathbf{z})$ as true data, *i.e.*, we want $D(G(\mathbf{z})) \approx 1$. In other words, for a given discriminator D, we update the parameters of the generator G to maximize the cross-entropy loss when $y = 0$, *i.e.*,

$$\max_G\{-(1-y)\log(1-D(G(\mathbf{z})))\} = \max_G\{-\log(1-D(G(\mathbf{z})))\}.$$

If the generator does a perfect job, then $D(\mathbf{x}') \approx 1$, so the above loss is near 0, which results in the gradients that are too small to make good progress for the discriminator. So commonly, we minimize the following loss:

$$\min_G\{-y\log(D(G(\mathbf{z})))\} = \min_G\{-\log(D(G(\mathbf{z})))\},$$

which is just feeding $\mathbf{x}' = G(\mathbf{z})$ into the discriminator but giving label $y = 1$.

To sum up, D and G are playing a "minimax" game with the comprehensive objective function:

$$\min_D \max_G\{-E_{x\sim\text{Data}}\log D(\mathbf{x}) - E_{z\sim\text{Noise}}\log(1-D(G(\mathbf{z})))\}.$$

3.7. Attention Mechanisms

3.8. Graph Neural Networks

3.9. Training losses

Chapter 4

Predictive Techniques for Scene Understanding

Avanzad, sin temor a la oscuridad.
Luchad jinetes de Theoden.
Caerán las lanzas, se quebrarán los escudos.
Aún restará la espada.
Rojo será el día, hasta el nacer del sol.
Cabalgad, cabalgad, cabalgad hacia la desolación
y el fin del mundo. Muerte, muerte, muerte.

Discurso de Theoden, Rey de Rohan
El Señor de los Anillos: El Retorno del Rey

4.1. SmartMOT

https://arxiv.org/pdf/2002.04849.pdf https://hal.science/hal-03347110/document https://www.mdpi.com/1424-8220/22/1/347

4.2. GAN based Vehicle Motion Prediction

4.3. Exploring Map Features

4.4. Leveraging traffic context via GNN

4.5. Improving efficiency of Vehicle Motion Prediction

Chapter 5

Applications in Autonomous Driving

La fuerza de tus convicciones determina tu éxito,
no el número de tus seguidores.

Reamus Lupin
Harry Potter y Las Reliquias de la Muerte, Parte 2

5.1. Motion Prediction Datasets

5.2. Multi-Object Tracking

5.3. Decision-Making

5.4. Holistic Simulation

Chapter 6

Conclusions and Future Works

El mundo no es todo alegría y color, es un lugar terrible y por muy duro que seas es capaz de arrodillarte a golpes y tenerte sometido a golpes permanente si no se lo impides; Ni tú ni yo ni nadie golpea mas fuerte que la vida. Pero no importa lo fuerte que golpeas, sino lo fuerte que pueden golpearte y los aguantas mientras avanzas, hay que soportar sin dejar de avanzar. ¡Así es como se gana!
Si tú sabes lo que vales, vé y consigue lo que mereces pero tendrás que soportar los golpes y no puedes estar diciendo que no estás donde querías llegar por culpa de él o de ella, eso lo hacen los cobardes y tú no lo eres.
TÚ ERES CAPAZ DE TODO.

Discurso de Rocky a su hijo
Rocky Balboa

6.1. Conclusions

In this thesis, a series of interaction-aware trajectory prediction methods, including single-agent trajectory prediction, multi-agent trajectory prediction, and multimodal trajectory prediction, are developed for autonomous driving. Besides, the impacts of trajectory prediction on trajectory planning are also investigated. In Chapter 3, a novel framework with consideration of vehicle-infrastructure heterogeneous interactions is proposed for trajectory prediction of a single target vehicle. In the proposed scheme, a heterogeneous graph is developed to represent the interactions, where the nodes contain features extracted from corresponding encoders. Besides, a novel heterogeneous graph social pooling (HGS) module is designed to extract high-level interaction features. The framework can be easily expanded for highway driving scenarios. Experimental results obtained using real-world driving datasets show that the proposed HGS method outperforms existing interaction-aware methods in terms of prediction accuracy. Besides, ablative studies demonstrate that the consideration of vehicle-infrastructure heterogeneous interactions effectively improves the prediction accuracy compared to those methods only considering inter-vehicle interactions. Then, the above prediction method for single-agent is generalized and expanded for heterogeneous multi-agent trajectory prediction in Chapter 4. To do this, a novel three-channel framework is designed to jointly consider traffc participants' dynamics, interaction, and map features. The driving scene is represented in a hybrid way, where the inter-agent interaction in the traffc system is represented with an edge-featured heterogeneous graph, and the shared local map is represented with a Bird's Eye View (BEV) image. Two shared Recurrent Neural Networks (RNNs) are adopted to capture vehicles' and pedestrians' dynamics features from their historical states, respectively. A novel heterogeneous edge-enhanced graph attention network (HEAT) is proposed to model the inter-agent interactions, and a map-sharing technique based on the gate mechanism is also leveraged to share the local map across all target agents. Experimental validations on real-world driving datasets of both urban and highway scenarios show that the proposed method not only achieves state-of-the-art performance but also can provide simultaneous predictions of multi-agent trajectories for a variable number of heterogeneous agents. Besides the unimodal predictions for single and multiple agents, this thesis also tackles the inherent multimodality problem of driving behaviors for prediction in Chapter 5. A novel map-adaptive multimodal trajectory prediction framework is proposed. Within this framework, through a single graph operation, a variable number of map-compliant trajectories and a non-map-compliant trajectory can be generated. Map-compliant predictions are conditioned on either a single candidate centerline (CCL) or a bunch of all CCLs, making the predictor adaptive to different road structures. The non-map-compliant prediction captures the irrational driving behavior for safety concerns. The driving scene is represented with a heterogeneous hierarchical graph containing both agents and their CCLs. A hierarchical graph operator (HGO) with an edge-masking technology is proposed to encode the driving scene. Validation on the Argoverse motion forecasting benchmark shows that the proposed method achieves state-of-the-art performance with the advantage of mapadaptive capacity. Beyond pure prediction, in Chapter 6, predictive planning and the impacts of prediction on downstream trajectory planning are also investigated. An interactionaware predictive planner, which is trained to imitate human driving behaviors, is designed to investigate the problem of how prediction would affect the performance of motion planning. The predictive planner is obtained by training an oracle planner, which is aware of target agents' ground truth future trajectories, and replacing the ground truth with the predicted trajectories for inference during implementation. Experimental results on a real-world dataset show that the proposed predictive planner achieves better performance over other baselines in terms of displacement error, miss rate, and collision rate. The gap between the predictive planner and the oracle planner shows that it is promising to further enhance the planning performance by improving the prediction accuracy. To be implemented in real-world self-driving systems, the proposed methods require upstream localization, perception, and tracking results, since the historical states of other traffc participants are needed as the input of the proposed methods. Perception can be realized using either or both of camera and LiDAR. Other sensors, such as radar, can also be used for better perception via sensor fusion. For the single-agent and multi-agent prediction methods in Chapter 3 and Chapter 4, we are using BEV maps, where only the map is needed. We do not need a BEV image to show the real-time tra!c. The map can be obtained from main-stream map providers and converted into images for the usage of our method. The mapadaptive multimodal method in Chapter 5, however, requires a high-definition map (HD map) of the local area since we need the candidate centerlines of vehicles of interest. The methods can run on both CPUs or GPUs, and using GPUs is suggested for faster inference.

6.2. Future Works

Although many studies have been done in trajectory prediction and path planning in the past years, there are still many aspects that need to be further investigated in the future. Scene representation and encoding. Researchers have proposed many methods to encode driving scenes with different representations. However, there is no unified representation of various driving scenes so far. The lack of a universal representation limits the generalizability of prediction methods with large-scale deployment in autonomous vehicles in the real world because a method can hardly be applied to a situation that cannot be described. Among many representation approaches proposed so far, graph-based representations are promising because a graph can accommodate an arbitrary number of heterogeneous objects and represent their interdependencies via directed edges. For example, when modeling a driving scene in the context of traffc systems, a node can represent a vehicle, a pedestrian, a lanelet, a junction, a traffc signal, etc. A new object can always be added to the existing graph. There are three important steps that need to be done to further improve the graph-based scene representation and encoding in the future. The first aspect is to construct the graph with proper connections, that is, to determine the edge set of the graph. This step needs to identify interdependencies between pairs of nodes and connect nodes with directed edges for information flow in the graph. Once the graph structure is settled, the second step is to assign the node, and edge features properly. This requires researchers to select or design proper encoders for different kinds of nodes and edges. Then the third step is required to design graph operators to handle the heterogeneity in the scene graphs. In this step, the advances in heterogeneous graph neural networks can be leveraged. Trajectory decoding. For future work, an immediate step is to generalize the map-adaptive multimodal prediction method proposed in Chapter 5 with uncertainty estimations. The uncertainty includes both motion and mode uncertainties. The motion uncertainty captures the distribution of agents' position over a planar map at each time step, and the mode uncertainty captures the possibility of driving modalities. The former can be modeled via bivariate Gaussian distributions, and the latter can be treated as a multi-class classification problem over a variable number of modalities. Then the next step can focus on generalizing uncertainty-aware multimodal predictions to multi-agent settings by modeling the joint distribution of multiple agents' behaviors. Social consistency should be considered in this step such that there is no conflict between any pair of trajectories in a joint modality in normal cases. Besides, trajectory predictors should be designed from the ego vehicle's point of view. One possible way is to design a decoder that can output trajectories upon the ego's request. For example, the predictor can focus on a small set of target agents requested by the ego vehicle rather than all the agents in sight. For a specific target agent, the predictor can focus on predicting its driving options that may affect the ego's planned trajectories. This attentive approach can reduce computation efforts for real-time implementations. Predictive planning. The ultimate goal of trajectory prediction is to further improve the performance of decision-making and motion control of autonomous vehicles with respect to safety, smartness, and effciency. So prediction must be integrated into the planning module, and therefore predictive planning is worthwhile exploring. There are many problems that should be addressed for the development of predictive planners. First, predictive planners should be able to address prediction uncertainty since prediction can never be exactly the same as the ground truth. Second, the relationship between prediction and planning needs to be further studied in order to answer the following questions: 1) How would the improvement in prediction affect the downstream planning performance? 2) Is there a floor of prediction error below which improving prediction accuracy leads to no improvement or even a negative effect on planning? 3) Can we design a predictive planner that is scalable to predictors of different uncertainties as long as these uncertainties are known? Third, learning-based motion planners should be further investigated since they have great potential to be incorporated with data-driven predictions. However, the current limitations in explainability and reliability need to be addressed. In general, plenty of effort is needed in these research areas.

Bibliography

[1] J. F. Arango, L. M. Bergasa, P. A. Revenga, *et al.*, “Drive-by-wire development process based on ros for an autonomous electric vehicle”, *Sensors*, vol. 20, no. 21, p. 6121, 2020.

[2] A. Ranga, F. Giruzzi, J. Bhanushali, *et al.*, “Vrunet: Multi-task learning model for intent prediction of vulnerable road users”, *arXiv preprint arXiv:2007.05397*, 2020.

[3] I. Gog, S. Kalra, P. Schafhalter, M. A. Wright, J. E. Gonzalez, and I. Stoica, “Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles”, in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 8806–8813.

[4] M.-F. Chang, J. Lambert, P. Sangkloy, *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.

[5] B. Wilson, W. Qi, T. Agarwal, *et al.*, “Argoverse 2: Next generation datasets for self-driving perception and forecasting”, *arXiv preprint arXiv:2301.00493*, 2023.

[6] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, “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.

[7] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, “Home: Heatmap output for future motion estimation”, in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 500–507.

[8] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, “Gohome: Graph-oriented heatmap output for future motion estimation”, in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 9107–9114.

[9] B. Varadarajan, A. Hefny, A. Srivastava, *et al.*, “Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction”, in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 7814–7821.

[10] M. Wang, X. Zhu, C. Yu, *et al.*, “Ganet: Goal area network for motion forecasting”, *arXiv preprint arXiv:2209.09723*, 2022.

[11] J. Schmidt, J. Jordan, F. Gritschneder, and K. Dietmayer, “Crat-pred: Vehicle trajectory prediction with crystal graph convolutional neural networks and multi-head self-attention”, in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 7799–7805.

[12] M. Liang, B. Yang, R. Hu, *et al.*, “Learning lane graph representations for motion forecasting”, in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, Springer, 2020, pp. 541–556.

[13] Z. Huang, H. Liu, J. Wu, and C. Lv, "Conditional predictive behavior planning with inverse reinforcement learning for human-like autonomous driving", *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[14] C. Tang and R. R. Salakhutdinov, "Multiple futures prediction", *Advances in neural information processing systems*, vol. 32, 2019.

[15] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2821–2830.

[16] S. Khandelwal, W. Qi, J. Singh, A. Hartnett, and D. Ramanan, "What-if motion prediction for autonomous driving", *arXiv preprint arXiv:2008.10587*, 2020.

[17] C. Tang, W. Zhan, and M. Tomizuka, "Interventional behavior prediction: Avoiding overly confident anticipation in interactive prediction", in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 11 409–11 415.

[18] H. Zhao, J. Gao, T. Lan, *et al.*, "Tnt: Target-driven trajectory prediction", in *Conference on Robot Learning*, PMLR, 2021, pp. 895–904.

[19] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving", *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[20] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "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.

[21] S. Casas, W. Luo, and R. Urtasun, "Intentnet: Learning to predict intention from raw sensor data", in *Conference on Robot Learning*, PMLR, 2018, pp. 947–956.

[22] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[23] J. Mercat, T. Gilles, N. El Zoghby, G. Sandou, D. Beauvois, and G. P. Gil, "Multi-head attention for multi-modal joint vehicle motion forecasting", in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 9638–9644.

[24] J. Gao, C. Sun, H. Zhao, *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. 11 525–11 533.

[25] J. Ngiam, V. Vasudevan, B. Caine, *et al.*, "Scene transformer: A unified architecture for predicting future trajectories of multiple agents", in *International Conference on Learning Representations*, 2022.

[26] S. Casas, A. Sadat, and R. Urtasun, "Mp3: A unified model to map, perceive, predict and plan", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 403–14 412.

[27] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "Covernet: Multimodal behavior prediction using trajectory sets", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 074–14 083.

[28] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 336–345.

[29] F. Marchetti, F. Becattini, L. Seidenari, and A. D. Bimbo, "Mantra: Memory augmented networks for multiple trajectory prediction", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7143–7152.

[30] Y. Biktairov, M. Stebelev, I. Rudenko, O. Shliazhko, and B. Yangel, "Prank: Motion prediction based on ranking", *Advances in neural information processing systems*, vol. 33, pp. 2553–2563, 2020.

[31] H. Cui, V. Radosavljevic, F.-C. Chou, *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.

[32] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction", *arXiv preprint arXiv:1910.05449*, 2019.

[33] T. Buhet, E. Wirbel, A. Bursuc, and X. Perrotton, "Plop: Probabilistic polynomial objects trajectory prediction for autonomous driving", in *Conference on Robot Learning*, PMLR, 2021, pp. 329–338.

[34] N. Rhinehart, K. M. Kitani, and P. Vernaza, "R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 772–788.

[35] H. Cui, T. Nguyen, F.-C. Chou, *et al.*, "Deep kinematic models for kinematically feasible vehicle trajectory predictions", in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 10 563–10 569.

[36] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking", in *2008 11th international conference on information fusion*, IEEE, 2008, pp. 1–6.