論文 / 著書情報
Article / Book Information

題目(和文)	自動運転のためのスマートモビリティデジタルツインに関する研究
Title(English)	Smart Mobility Digital Twin for Autonomous Driving
著者(和文)	WANG KUI
Author(English)	Kui Wang
出典(和文)	学位:博士(学術), 学位授与機関:東京科学大学, 報告番号:甲第378号, 授与年月日:2025年3月26日, 学位の種別:課程博士, 審査員:阪口 啓,廣川 二郎,小竹 元基,TRAN GIA KHANH,佐藤 育郎,新熊 亮一
Citation(English)	Degree:Doctor (Academic), Conferring organization: Institute of Science Tokyo, Report number:甲第378号, Conferred date:2025/3/26, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Doctoral Dissertation

Smart Mobility Digital Twin for Autonomous Driving

Supervisor Professor Kei Sakaguchi

Department of Electrical and Electronic Engineering
Graduate School of Engineering
Institute of Science Tokyo

Kui Wang

Contents

List of Figures

List of Tables

Acknowledgments

Since I came to Japan in 2022 to pursue my doctoral studies, three years have passed in the blink of an eye. During this journey, my deepest gratitude goes to my Ph.D. supervisor, Professor Kei Sakaguchi. His invaluable guidance and support have been the cornerstone of my successful completion of the doctoral program. Moreover, his dedication to work and unwavering passion for academia have profoundly inspired and influenced me, leaving a lasting impact on my future career choices and development.

I would also like to extend my heartfelt thanks to other professors who have supported me during my doctoral studies, including Professor Walid Saad, with whom I collaborated on the JUNO research project, as well as Professor Yonghui Li and Professor Changyang She from the University of Sydney, where I had the privilege of engaging in an exchange program. I am also grateful to Associate Professor Tao Yu and Assistant Professor Zongdian Li, senior members of my laboratory, for their constant support and mentorship. Without their selfless help and encouragement, I would not have successfully completed my doctoral research. I also wish to thank the reviewers of my thesis, i.e., Professor Shinkuma, Professor Sato, Professor Shino, Professor Hirokawa, and Associate Prof. Tran—whose professional and insightful feedback during the defense process greatly improved my thesis. Their guidance has not only helped me identify the shortcomings in my research but also clarified the directions in which I need to further develop professionally.

Finally, and most importantly, I owe my deepest gratitude to my family for their unwavering support, especially my wife and my mother. My wife, who is also pursuing her studies in Japan, has taken care of me with unparalleled care and love throughout this journey. Additionally, my mother has always provided steadfast financial support and invaluable advice on life matters, just as she has done for me over the past three decades. Their unconditional love and support have been my greatest source of strength and motivation.

Abstract

Over the past two decades, digital twins (DTs), have driven significant advancements across various industrial domains, including manufacturing, aerospace, and healthcare. With the rapid development of autonomous driving technologies and vehicle-to-everything (V2X) communications, integrating DTs into vehicular platforms is anticipated to further revolutionize smart mobility systems by enhancing connectivity, efficiency, and safety on the roads.

This dissertation introduces a novel smart mobility DT (SMDT) platform for the control of connected and automated vehicles (CAVs) over next-generation wireless networks. The SMDT platform is implemented with state-of-the-art (SOTA) products, e.g., CAVs and roadside units (RSUs), and emerging technologies, e.g., cloud and cellular V2X (C-V2X). As a result, The SMDT creates a comprehensive and dynamic virtual representation of the physical transportation environment, enabling real-time data exchange and interaction between vehicles RSUs, and central cloud.

The platform is applied to develop a global DT-based CAV navigation system aimed at improving traffic efficiency. By optimizing vehicle route planning through real-time data analysis on the cloud server, the system allows CAVs to make more informed routing decisions. Proof-of-concept (PoC) experiments are conducted to validate system performance. The performance of SMDT-based CAV navigation system is evaluated from two standpoints: *(i)* the rewards of the proposed navigation system on traffic efficiency and safety and, *(ii)* the latency and reliability of the SMDT platform. Our experimental results using SUMO-based large-scale traffic simulations show that the proposed SMDT can reduce the average travel time and the blocking probability due to unexpected traffic incidents. Furthermore, the results record a peak overall latency for DT modeling and route planning services to be 155.15 ms and 810.59 ms, respectively, which validates that our proposed design aligns with the 3GPP requirements for emerging V2X use cases and fulfills the targets of the proposed design.

The SMDT platform is also applied to a local DT (LDT)-assisted hybrid autonomous driving system for improving safety and efficiency in traffic intersections. By leveraging RSUs equipped with sensory and computing capabilities, the proposed system continuously monitors traffic, extracts human driving knowledge, and generates intersection-specific local driving agents through an offline reinforcement learning (RL) framework. When CAVs pass through RSU-equipped intersections, RSUs can provide local agents to support safe and efficient driving in local areas. Meanwhile, they provide real-time cooperative perception to broaden onboard sensory horizons. Hardware-in-the-loop (HiL) simulations and PoC tests validate system performance from two standpoints: *(i)* The peak latency for cooperative perception and local agent downloading are 8.51 ms and 146 ms, respectively, aligning with 3GPP requirements for V2X and model transfer use cases. Moreover, *(ii)* local driving agents can improve safety measures by 10% and reduce travel time by 15% compared with conventional onboard systems. The implemented prototype also demonstrates reliable real-time performance, also fulfilling the targets of the proposed system design.

Chapter 1

Introduction

1.1 Background

1.1.1 Autonomous Driving: Past, Present, and Future

Autonomous driving system (ADS) has evolved significantly since its inception, marking a transformative journey from conceptual theories to practical implementations and commercial practices. The quest for self-driving vehicles began in the mid-20th century, aiming to enhance transportation efficiency and safety. Early experiments in the 1980s and 1990s, such as Carnegie Mellon University's NavLab and the European PROMETHEUS project (see [1] and [2]), laid the groundwork by integrating basic sensors and rudimentary algorithms to enable limited self-driving capabilities.

Since the beginning of the 21st century, we have witnessed a surge in the development of the ADS field, propelled by substantial investments from both public and private sectors. The Defense Advanced Research Projects Agency (DARPA) organized the Grand Challenge competitions in 2004 and 2005, pivotal in advancing autonomous vehicle technologies by challenging teams to navigate automated vehicles to finish the 150-mile off-road parkour [3], where all attendees failed in 2004 and only five teams completed the track without any human interference in 2005. Initiated in 2009, Google's autonomous vehicle project made significant strides by employing LiDAR, radar, and advanced machine learning (ML) algorithms to navigate complex urban environments [4]. Since its initial launch in 2014, the Society of Automotive Engineers (SAE) has defined levels of driving automation, from Level 0 (no driving automation) to Level 5 (full driving automation) in the context of motor vehicles

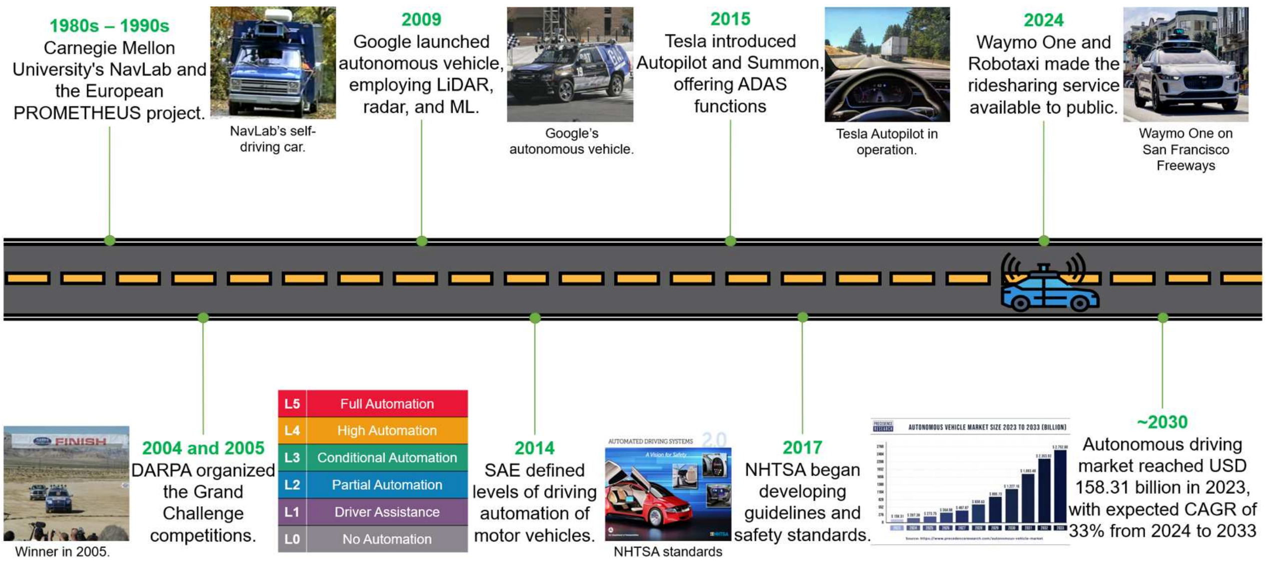


Figure 1.1: Past, present, and future of autonomous driving.

and their operation on roadways [5], which has been the industry's most-cited source for driving automation. Tesla introduced Autopilot and Summon in 2015, offering advanced driver assistant systems (ADAS) and paving the way toward full autonomy [6]. Unfortunately, fatal accidents involving Tesla cars operating on Autopilot kept growing since 2016 [7]. Hence, Regulatory bodies like the National Highway Traffic Safety Administration (NHTSA) began developing guidelines and safety standards to govern the testing and deployment of ADSs, acknowledging the technology's rapid advancement and potential impact on transportation (e.g. [8]).

At present, autonomous driving has transitioned from theoretical constructs to practical implementations, with numerous companies and research institutions developing vehicles that can operate under specific conditions. Some novel commercial attempts in terms of autonomous ridesharing, e.g. Waymo One in Los Angeles, US [9], and Robotaxi in Wuhan, China [10], have also significantly contributed to the flourishing development of the autonomous driving field. A recent report indicates that the global autonomous vehicle market size was estimated at USD 158.31 billion in 2023 and is predicted to hit around USD 2,752.80 billion by 2033, poised to grow at a compound annual growth rate (CAGR) of 33% from 2024 to 2033 [11]. This remarkable market performance highlights the crucial role of autonomous driving as a cornerstone in the future of human society. In Figure 1.1, we summarize the timeline of ADS development along with some key milestones.

1.1.2 System Components and Architecture of Autonomous Driving

The development of autonomous driving is characterized by foundational research and incremental technological advancements. Pioneering efforts focused on developing core components like computer vision, sensor fusion, and control systems. However, these early-stage research were constrained by computational limitations and lacked the sophistication required for real-world application. The primary objective during this period was to prove the feasibility of autonomous vehicles and to understand the complexities involved in mimicking human driving behavior. In recent years, propelled by vast advancements in sensing (e.g., 3D LiDAR and event camera), computing (e.g., ML, artificial intelligence (AI), and hardware acceleration), and communication technologies (e.g., millimeter wave (mmWave), dedicated and short-range communications (DSRC)), autonomous driving techniques are advancing towards higher intelligence, connectivity, and generalization, thereby gradually evolving into various branches of approaches. ADSs can be categorized based on their algorithmic design and connectivity paradigms. From an algorithmic standpoint, they are classified into modular systems, as shown in Figure 1.2, and end-to-end (E2E) driving systems, as shown in Figure 1.3. Regarding connectivity, we divide them into ego-only systems, cooperative systems, and cyber-physical systems (CPS)-based systems. Figure 1.5 and Figure 1.6 show general system architectures of cooperative and CPS-based systems.

1.1.2.1 Modular Systems

Modular ADSs are structured as pipelines where the driving task is decomposed into distinct and sequential modules, typically including *navigation, environmental perception, localization, prediction, planning, and control*, as shown in Figure 1.2. Each module processes specific inputs and produces outputs for the subsequent module. For instance, sensor data are first interpreted by the perception module to detect objects and understand the environment. The localization module will then determine the vehicle's position within a high-definition (HD) map. Subsequent modules predict the behavior of surrounding entities, plan a trajectory, and execute control commands to actuate the vehicle [12].

The modular approach leverages established research in robotics, computer vision, and control systems, and benefits from the ability to optimize and troubleshoot individual components independently, facilitating the integration of cutting-edge techniques within each

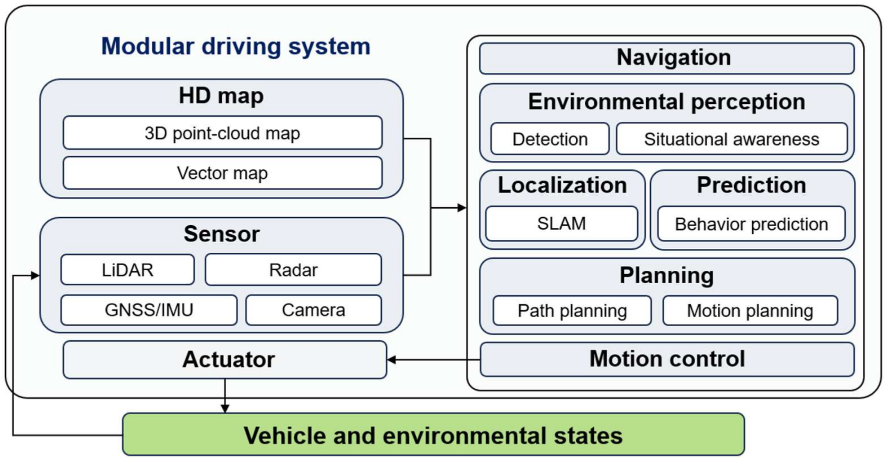


Figure 1.2: Functional block diagrams of a generic modular system.

module. However, a significant drawback is the potential for *error propagation* [13]. Mistakes or uncertainties in one module can cascade through the pipeline, adversely affecting the overall system performance. In addition, the complexity of integrating multiple modules also poses challenges in parameter tuning and system testing.

1.1.2.2 E2E driving systems

The E2E approach adopts a holistic approach by using supervised deep learning (DL) [14], deep reinforcement learning (DRL) [15], or the more recent large language vision models (LLVM) [16] to map raw sensor inputs directly to control commands without explicitly defined intermediate representations, as shown in Figure 1.3. These systems leverage large datasets and deep neural networks (DNN) to learn driving policies that encapsulate perception, decision-making, and control within a single model.

The E2E driving simplifies the architecture by eliminating hand-crafted modules, potentially enabling the system to discover optimal features and representations autonomously, and making it more adaptable to complex and unstructured environments. However, it raises concerns regarding *interpretability and transparency*, as the internal decision-making process of DNN is a blackbox, which is hard to understand, explain, or validate, especially in safety-critical applications. Additionally, the *curse of rarity (CoR)* problem, i.e., the occurrence probability for the safety-critical event being so rare that most available data contain very little information about the rare events, also presents significant challenges in ensuring the safety of autonomous vehicles using E2E models [17].

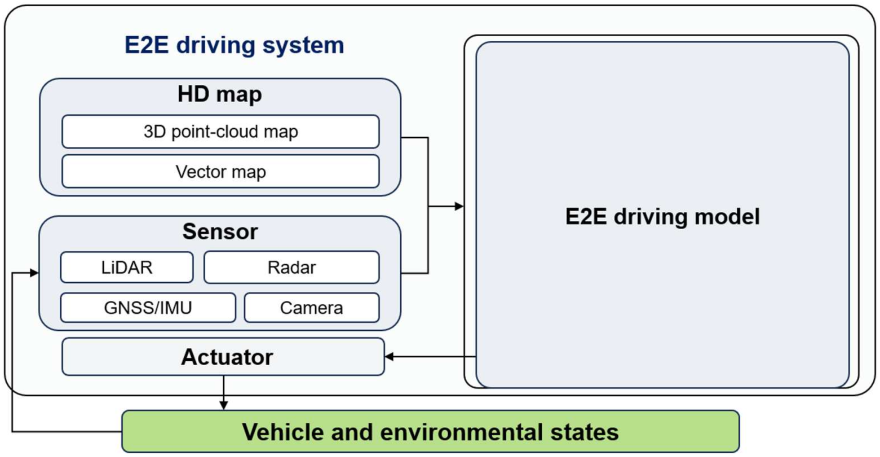


Figure 1.3: Functional block diagrams of an E2E driving system.

1.1.2.3 Ego-only systems

Ego-only systems operate solely relying on onboard sensors and processing capabilities to perceive the environment, make decisions, and control the vehicle. These systems are designed to be self-sufficient, handling all necessary functions without external inputs. The autonomy of ego-only systems simplifies deployment by avoiding dependencies on external infrastructure or communication networks. However, they are limited by the sensing range and computational resources available on the vehicle, which can impact their ability to handle complex scenarios, such as blind-spot obstacles or rapidly changing environments.

1.1.2.4 Cooperative Systems

The cooperative way extends the capabilities of individual ADS by enabling communication and collaboration between vehicles and other smart entities through vehicle-to-everything (V2X) communications, mainly incorporating three types of interfaces: vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-network (V2N) [18]. The connectivity among such smart entities or devices facilitates *cooperative perception and cooperative driving* to enhance the safety and efficiency of connected and automated vehicles (CAVs) and transportation systems, as shown in Figure 1.4.

Cooperative perception allows vehicles to share different levels of sensory information, such as raw sensor data, detected objects, or environmental conditions, to build a more comprehensive understanding of the surroundings [19]. For example, a vehicle may receive alerts about hazards beyond its sensory horizon, such as sudden braking of a vehicle several

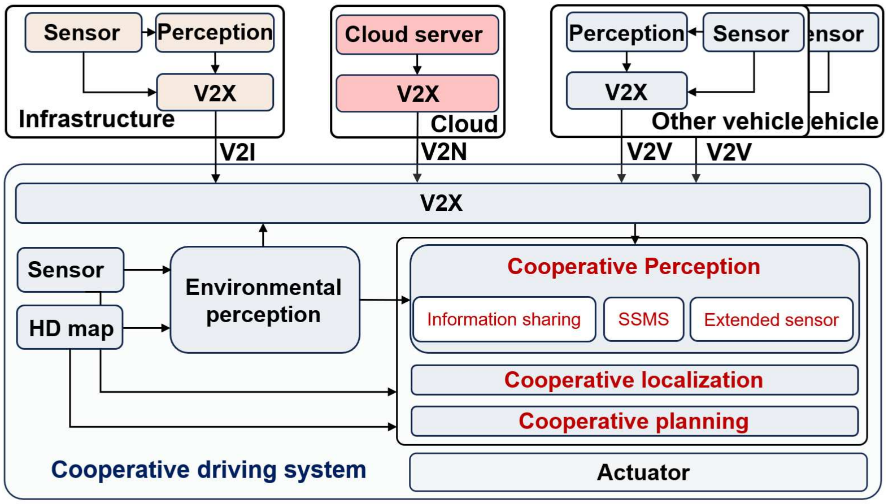


Figure 1.4: Functional block diagrams of a cooperative ADS.

cars ahead or adverse road conditions reported by other vehicles. This shared awareness can significantly improve reaction times and decision-making accuracy.

On the other hand, *cooperative driving* involves the coordination of maneuvers among multiple vehicles to optimize traffic flow and enhance safety [20]. By sharing intentions, planned trajectories, or motion states, vehicles can collaboratively execute tasks such as lane changes, ramp merging, or intersection crossing. Maneuvering level of coordination reduces the risk of collisions, minimizes traffic congestion, and improves overall transportation efficiency. For instance, vehicles approaching a congested intersection can negotiate right-of-way in real time, or a group of vehicles traveling on a highway can form a platoon to reduce air resistance and improve fuel efficiency.

While cooperative systems offer substantial benefits, they also introduce challenges related to communication reliability, network security, and standardization. The dependency on V2X communication necessitates robust protocols to handle latency, data loss, and potential security threats, such as spoofing or hacking. Hence, some organizations for standardization, such as the European Telecommunications Standards Institute (ETSI) and the Third Generation Partnership Project (3GPP), have proposed various technical reports (TR), i.e. TR 102 638 [21] and TR 22.885 [22], to introduce a rich set of V2X applications, including cooperative adaptive cruise control (CACC), cooperative lane change (CLC), cooperative collision avoidance (CoCA), etc. These efforts underline the critical need for a unified framework to ensure the seamless integration and secure operation of V2X-enabled cooperative systems,

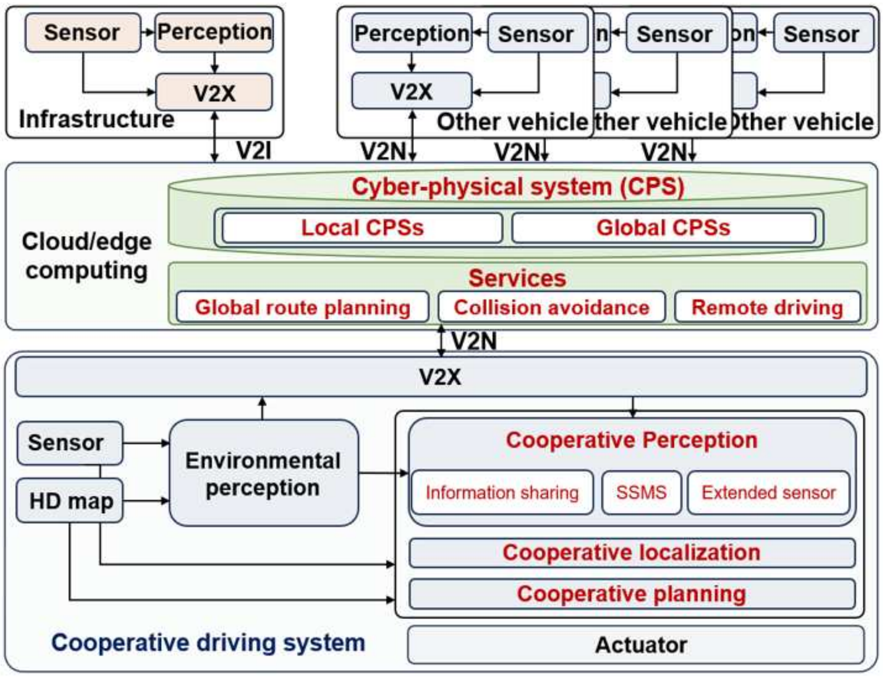


Figure 1.5: Functional block diagrams of a CPS-based ADS.

paving the way for the realization of the full potential of cooperative systems in enhancing traffic efficiency and safety.

1.1.2.5 CPS-based systems

In more recent years, the appearance of CPS has brought a transformative evolution in the autonomous driving field, by integrating sensing, computation, networking, and physical processes to create an intelligent and interconnected environment. In CPS, cyberspace with embedded computers and networks can keep monitoring and controlling physical entities through feedback loops, where physical processes affect cyberspace and vice versa [23]. In the context of autonomous driving, the CPS-based approach seamlessly merges cyber components (e.g. algorithms, data analytics, and communication networks) with physical elements (e.g. sensors, actuators, vehicles, and infrastructure), which facilitates real-time data exchange and advanced control strategies, enhancing capabilities beyond those of ego-only or cooperative systems [24].

As illustrated in Figure 1.5, By leveraging cloud/edge computing and V2X communications, the hierarchical structure of CPS ensures system scalability and reliability, with local systems focusing on delay-sensitive operational tasks while global systems optimize overall

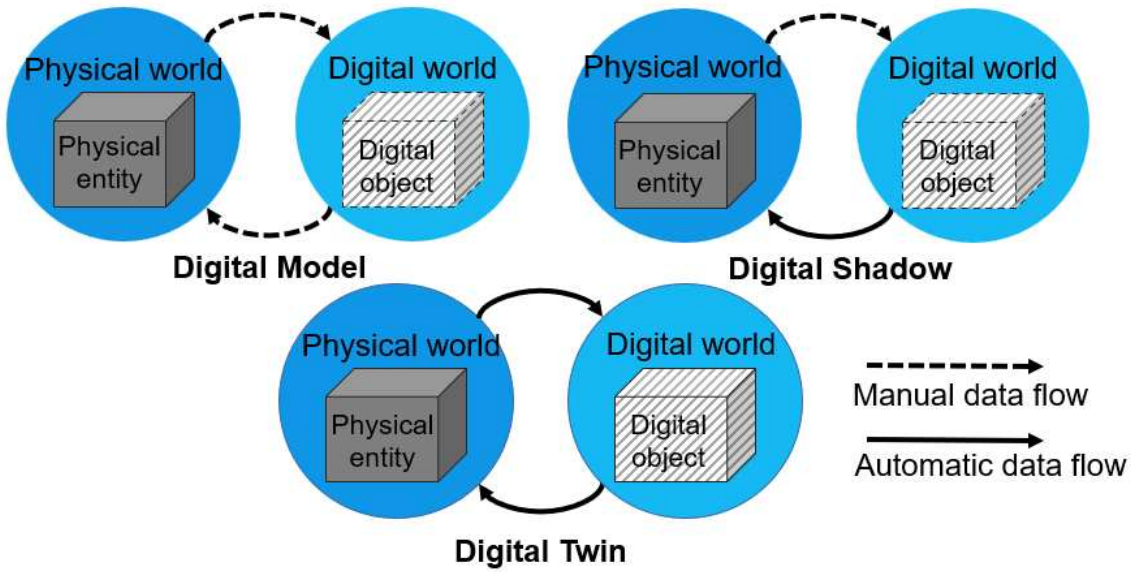


Figure 1.6: Data flow in digital model, digital shadow, and DT.

traffic flow and safety. The importance of such technology is that it can facilitate a spectrum of new derivative services, such as global route planning, collision avoidance, dynamic map generation, and remote driving to further enhance the robustness, safety, and efficiency of ADS, paving the way for smarter intelligent transportation systems (ITS).

However, CPS-based approach also introduces great challenges related to complex system integration, network security, and data privacy. Developing these systems requires integrating various technologies and establishing standards for data formats, communication protocols, and system interfaces [25]. The extensive connectivity increases vulnerability to cyber attacks, necessitating robust cybersecurity measures to protect user privacy against malicious interference [26].

1.1.3 Digital Twin: Past, Present, and Future

The digital twin (DT), as an emerging representation of CPS, attracted increasing attention over the past decade. Although the definitions of the DT vary in different versions, the basic concepts are essentially the same: A DT is a digital replica of a living or nonliving physical entity. As shown in Figure 1.6, we emphasize and distinguish the definitions of *digital model, digital shadow, and DT*, highlighting that a DT requires *automatic bidirectional data flow* between the physical and digital worlds [27].

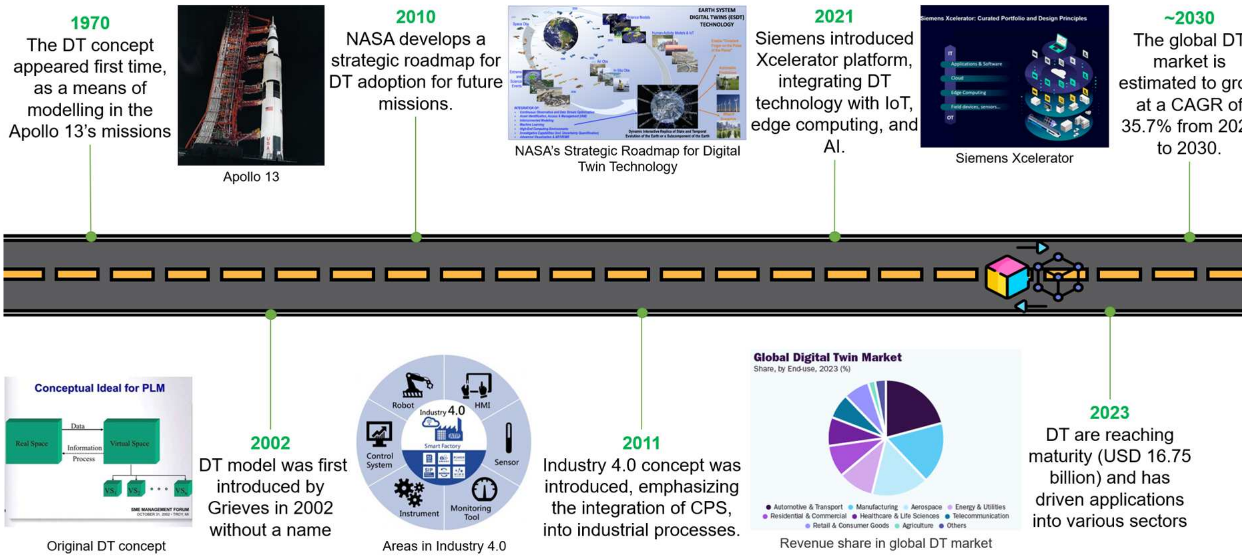


Figure 1.7: Past, present, and future of DT.

The origin of DT can be traced back to 1970 when the National Aeronautics and Space Administration (NASA) utilized paired simulations to mirror spacecraft systems during the rescue of Apollo 13. In 2002, Dr. Michael Grieves formally introduced the DT concept during a presentation on product lifecycle management (PLM) at the University of Michigan. He proposed a virtual representation of physical products to enhance manufacturing processes and lifecycle management [28], where the framework comprised three primary components: the physical product, its virtual counterpart, and the information flow between them.

The term “digital twin” appeared for the first time in 2010 when NASA adopted it as a key element in its 2010 technology roadmap, marking a shift of DT from theoretical discussions to practical implementation. With the realm of the Internet of Things (IoT) rapidly expanding beyond the conventional frameworks after the introduction of Industry 4.0 in 2011, DT technology has begun to establish its foothold across different industries, such as Microsoft Azure [29], Siemens Xcelerator [30], NVIDIA Omniverse [31], and even city-wide DT like Tokyo DT project [32].

At the forefront of these viable sectors, integrating DTs into the transportation and traffic fields constitutes a promising solution for critical challenges like traffic congestion and road accidents [33]. A recent report indicates that the global DT market reached USD 16.75 billion in 2023 and was predicted to grow at a CAGR of 35.7% until 2030 [34]. Notably, the automotive and transport segment dominated the DT market in 2023, accounting for over

21.0% of its total revenue. This remarkable market performance highlights the crucial role of DTs as a cornerstone in the smart mobility infrastructure. In Figure 1.7, we show the timeline of the development of DT along with some key milestones.

1.1.4 Convergence of DT and Autonomous Driving

Based on the discussion above, we believe that both the autonomous driving and DT domains have reached sufficient maturity for integration, presenting an exciting new research frontier. This convergence is driven by the overlapping objectives of these fields: enhancing system safety, efficiency, reliability, and scalability while addressing the complexities of real-world applications. The integration of DT into autonomous driving introduces several benefits as follows:

- *Cloud/Edge Resources:* The convergence of DT and autonomous driving leverages *cloud and edge computing* to enable efficient data processing and resource allocation. Cloud computing provides high scalability and computing power, while edge computing ensures low-latency responses by processing time-critical tasks locally, such as safety-critical maneuvers. This hybrid approach supports real-time synchronization between physical and digital entities, paving the way for advanced autonomous driving applications.

- *Traffic efficiency improvement:* DT integrates *global traffic information* from various sources, including vehicles, infrastructure, and historical data, enabling the optimization of traffic flow on a systemic level. By making real-time, data-driven decisions such as adjusting traffic signals, vehicle routing, and platooning, DT enhances the overall efficiency of transportation networks. This holistic optimization minimizes congestion, reduces travel times, and improves the utilization of road infrastructure.

- *Road safety enhancement:* Distributed intelligence within DT, particularly at the edge (e.g., smart infrastructure), facilitates *vehicle-road collaboration*, thereby enabling proactive safety services for autonomous vehicles. Real-time data analysis and decision-making at the edge enable DT to provide timely alerts about potential hazards, such as pedestrian crossing or sudden braking events. This capability significantly enhances situational awareness, ensuring safer navigation for autonomous vehicles even in complex and safety-critical environments.

- *Big data from ITS:* The integration of DT with ITS generates vast amounts of data from CAVs, infrastructure, and multiple sensors. This big data is instrumental in addressing challenges such as the CoR in autonomous driving. By providing a continuous stream of real-world data, DT can continuously enhance the robustness of AI/ML models and improve the reliability of autonomous systems.

- *Driving world model:* A world model represents an agent's understanding of its environment, incorporating dynamic and causal relationships to predict outcomes and inform decisions. In the context of DT, the driving world model is enriched by causal inference and real-time data, allowing autonomous systems to anticipate future events and adapt their strategies accordingly. For example, a DT-enhanced world model could predict how traffic patterns might evolve in response to road closures or weather changes, enabling more effective and reliable decision-making.

- *DT-Based simulation for testing and validation:* DT provides a high-fidelity virtual testing environment for autonomous driving systems, allowing for extensive validation under diverse scenarios without the risks and costs of physical testing. DT-based simulations can replicate real-world conditions, integrate real-time data, and simulate rare or extreme events, such as sudden mechanical failures or extreme weather conditions. These capabilities accelerate the development cycle while ensuring the safety and robustness of autonomous driving algorithms.

- *Personalized mobility services:* The integration of DT enables personalized mobility solutions tailored to individual user preferences and requirements. By analyzing user behavior, preferences, and travel patterns, DT can provide customized services such as adaptive ride-sharing, optimized routing, and user-specific vehicle configurations.

1.2 Research Objectives and Related Works

This thesis presents the author's efforts to build the smart mobility digital twin (SMDT) for traffic efficiency improvement and road safety enhancement, by introducing *cloud-based services* and *vehicle-road collaboration* at the edge (i.e., roadside units (RSUs) and CAVs). The motivation for studying SMDT can be concluded into three points:

- *Resource allocation:* Real-time data processing is critical for autonomous driving, where delays in computing or communication can lead to unsafe decisions, especially in dynamic scenarios like intersections. If computational and communication resources from cloud/edge systems are effectively coordinated within SMDT, it will become possible to provide timely and reliable support for CAVs, enhancing road safety and operational efficiency.

- *Traffic management:* Navigation (i.e., global route planning) remains a persistent challenge for autonomous driving. The sensory horizon of CAVs is inherently limited, and even with cooperative perception, CAVs often lack sufficient real-time traffic information to perform optimal global route planning. If comprehensive traffic data, integrating real-time inputs from multiple sources, can be dynamically processed and utilized in SMDT, it could enable accurate traffic analysis and optimal route planning for CAVs

- *Safety enhancement:* As discussed in Section 1.1.2, the scarcity of safety-critical events in datasets (i.e., CoR problem) poses a significant obstacle to the reliability of AI/ML-based ADS. Once we enable SMDT with the capability of continuous traffic monitoring, some knowledge of driving behaviors, including extreme and rare scenarios, can be accumulated and applied, thereby ensuring safer and more adaptive decision-making in complex environments.

However, existing mobility DTs and vehicle-road collaboration technologies have considerable limitations, which hinder the establishment of SMDT. Thus we need to consider a new design.

1.2.1 Limitations of Existing Mobility DTs

Given the benefits of DT technology, researchers have made some efforts to introduce the concept of DTs to vehicular technologies in recent years [35,36,37]. To build a real-time DT model of traffic environments and provide feedback services for CAVs, it is important to handle three technical challenges: sensing, communication, and computation. Sensing plays a pivotal role in environmental information capture, which is integral for the accurate representation and predictive analysis of complex vehicular and traffic systems. The work in [38] summarizes different ways for traffic information acquisition and traffic congestion monitoring, such as using stationary roadside sensors, probe-vehicle-based techniques, vehicular networks, and social networks. Among these various methods, stationary roadside sensors, typically referred to as RSUs, can accurately and continuously monitor the traffic condition and traffic flow [39,40]. Although the construction of RSUs is currently expensive due to the inclusion of some specialized and high-performance hardware like LiDARs, such cost can significantly decrease while the hardware becomes mass-market and widely adopted in autonomous driving and infrastructures of smart cities [41]. Hence, the use of sensor-equipped RSUs to establish a traffic DT has become popular in some related research. For example, in [42], multiple sensors deployed in the smart infrastructure are used to detect pedestrians and realize object-level central perception on the cloud. The authors in [43] develop a 3D DT framework of an ITS based on roadside sensing methods. However, it is impractical to anticipate that the sensing range of RSUs can totally cover the entire traffic [44]. Consequently, a more realistic method is to enhance the utilization of vehicular onboard sensors, making CAVs also contribute to the modeling of DT, as discussed in [45,46,47].

When it comes to the communications aspect, the concept of V2X provides a collaborative approach where multiple smart entities can cooperatively perceive the traffic environment [48,49]. Hence, V2X has found a multitude of DT-like or DT-related applications within the field of autonomous driving and ADAS. The authors in [50] develop a software-defined network (SDN) to achieve global path planning for connected vehicles, based on vehicle-to-BS and vehicle-to-UAV communications. In [51], authors establish a V2I network between road vehicles and traffic lights to build a traffic light DT and optimize traffic control. In [52], a software-defined vehicular network (SDVN) architecture for heterogeneous V2X is designed to realize cooperative perception and ensure safety in autonomous driving. Fundamentally, a DT represents a form of central cooperative perception, wherein sensory data is aggregated and fused at a central server. Thus, the establishment of a V2X network lays the groundwork for the extrapolation of mobility DT platforms.

In aligning with the discussed sensing and communication challenges, computation also emerges as a critical component in the DT frameworks, especially the collaboration of cloud and edge computing [53]. Since edge computing is well-suited for handling delay-sensitive tasks and can also offload the cloud computation, it is often employed in some related research for functions like environmental perception and object detection [42], as well as autonomous driving system tasks like motion planning and motion control [46]. On the other hand, the central cloud, as a data pool and high-level decision-maker, aggregates data from various smart entities and makes decisions, then provides feedback or services back to smart entities. For example, [54] demonstrates the use of a first-in-first-out slot reservation algorithm within a centralized server for cooperative driving at non-signalized intersections. In another instance, [55] presents a cloud control testbed for validating multi-vehicle cooperation and vehicle-road-cloud integration. Hence, in the proposed SMDT platform, the synergetic integration of cloud and edge computing necessitates careful consideration.

In Table 1.1, we summarize some related works and compare the main features of their research [35,36,42,43,46,50,51,54,55], where the level of driving automation proposed by SAE is applied to assess the degree of autonomy in these studies. To our knowledge, there are few studies investigating the confluence of DT technology and autonomous driving. Additionally, these studies predominantly employ validation methods of simulation or field trials. Simulation-based research not only establishes DT models but can also derive a series of new services. However, those conducting field trials face challenges in demonstrating the feedback

Table 1.1: Related works of Mobility DT

Ref.	Key components: Cloud	Vehicles	RSUs	SAE level	DT functions: Modeling	Service	Validation methods
	Objectives (use cases)						
[35]	√	√	√	L2	√	√	Field test&Simulation
	Personalized adaptive cruise control (P-ACC).						
[42]	√	√	√	L2	√		Field trial
	Cooperative perception to create traffic DT.						
[43]	√		√		√		Field trial
	Cooperative vehicle-infrastructure system.						
[46]		√		L4	√		Field trial
	Validation framework for autonomous driving.						
[50]	√	√				√	Simulation
	Path planning considering unexpected events.						
[51]	√	√	√		√	√	Simulation
	Traffic light control system.						
[54]	√	√		L2	√	√	Simulation
	Cooperative driving at intersections.						
[55]	√	√			√		Sandbox
	Validation of multi-vehicle cooperation.						

from the digital to the physical world, where the concept of DT is conceived as a simulation environment for modeling real-world entities with high fidelity. In our perspective, a DT transcends mere mapping and reconstruction of the physical space within cyberspace. As shown in 1.6, it should also enable an automatic bidirectional data flow between the physical and digital worlds to fully unlock the potential of DTs. i.e., leveraging global information in DT to optimize and control physical entities.

1.2.2 Limitations of Current Vehicle-Road Collaborations

In alignment with cooperative ADS discussed in Section 1.1.2, existing vehicle-road collaboration technologies can be also broadly categorized into cooperative perception and cooperative driving. Cooperative perception allows a network to leverage advancements in ITS and V2X technologies to improve the situational awareness capability of CAVs by reducing the occurrence of safety-critical events [17]. For instance, CAVs can obtain more road traffic information via V2I communications which allows them to make decisions more reasonably. Hence, various studies have focused on enhancing CAV operations by using cooperative perception [56,57,58,59,60,61,62,63,64,65,66,67]. For example, the work in [56] proposes raw sensor data sharing to support automated driving via mmWave V2X. The authors in [57] propose a birds-eye-view (BEV) grid occupancy prediction method to support various applications like driving safety warnings and cooperative traffic control. Moreover, the work in [58] and [59] notices the importance of unique features of different intersections, where intersection-specific strategies are deployed for driver-intended movements prediction and road-to-vehicle visual information sharing, respectively. Several works such as [60] and [61] also highlight the practical aspects of cooperative perception system implementation and deployment.

On the other hand, cooperative driving allows CAVs to coordinate their maneuvers in complex traffic environments, such as intersections and highways, thereby improving overall traffic safety and efficiency. Research in this area mainly focuses on developing sophisticated algorithms and robust communications protocols that enhance the collaborative decision-making capabilities of CAVs in dynamic and uncertain environments [70,73,74,75,76,68,69, 77,78,79]. For instance, multi-vehicle coordination systems utilize real-time data sharing and predictive modeling to enable vehicles to anticipate the actions of others and adjust their trajectories accordingly, thereby reducing the risk of collisions and improving traffic flow [70,73,74]. Advanced intersection management models integrate traffic signal optimization and vehicle trajectory planning, considering factors such as speed, density, and pedestrian movements, to minimize congestion and enhance safety [75,76,68,69].

In recent years, the introduction of DT in the transportation systems has also become a hot topic in cooperative perception and driving fields as the synergistic integration between the physical and cyber worlds allows for enhanced real-time decision-making and predictive

Table 1.2: Related works of vehicle-road collaboration

Ref.	**Key components: Cloud**	**Key components: Vehicles**	**Key components: RSUs**	**SAE level**	**Unique traffic features**	**Validation methods**	**Objectives (use cases)**
Cooperative perception-related studies							
[57]		√	√	-		Simulation	Generation of future BEV occupancy map.
[58]		√	√	-	√	Dataset	Early prediction of intersection movement and warning.
[59]		√	√	-	√	Dataset	Adaptive construction of a dynamic perception representation.
[67]		√	√	-		Simulation	Intersection traffic safety framework for collision avoidance.
Cooperative driving-related studies							
[68]		√	√	-		Simulation	Federated learning framework to reduce idling time.
[69]			√	-		Simulation	Traffic signal control system to reduce idling time.
[70]		√	√	L4/L5		Indoor robots	Intersection coordination framework for trajectory planning.
DT-based studies							
[71]		√	√	-		Field test	Highway DT for CAV perception enhancement.
[72]	√	√		L2		Field test	DT-based ramp merging framework.
[54]	√	√		L2		Simulation	DT-assisted FIFO framework at intersections

analytics [54,80,81,71,82,72]. For example, in [82] and [72], authors propose and implement transportation DT systems to improve automated driving functions like environmental perception. The work in [54] focuses on the DT-based first-in-first-out (FIFO) slot reservation algorithm in traffic intersections to reduce travel time and energy consumption.

However, most existing research in cooperative perception and driving fields primarily focuses on the sharing of dynamic information, such as vehicle states, predicted behaviors, and decisions [54,57,58,59,67,70,68,69,71,72], as shown in Table 1.2. Indeed, dynamic information is crucial for operational safety, but introducing *(transient) static information and inherent unique road traffic features*, and based on which, generating *local strategies* have great potential to transformatively boost the development of autonomous driving. For example, a human driver commuting daily past an intersection near a kindergarten will quickly learn to pay extra attention and adopt a more careful and conservative *local strategy*. How to enable CAVs to gain such knowledge and *local strategies* is another core motivation of this thesis. Additionally, most related studies employ validation methods of simulation or collected datasets, lacking implementation and outdoor tests to validate the system effectiveness in the real world. Hence, validating the benefits of *local strategy* with real-world experiments is also a key consideration.

1.3 Summary of Contributions

To validate the feasibility of applying the SMDT platform within the domain of autonomous driving and explore its potential, the most important task is the system design and implementation in the real world, which poses numerous challenges. Firstly, it is essential to create a real-time virtual representation of real-world traffic environments (i.e., traffic DT), which involves sensing, perceiving, and visualizing various static and dynamic entities. Secondly, the system design should consider safety and robustness issues, because of massive data collection from traffic environments and the requirement for autonomous vehicles to operate in unpredictable conditions, which necessitates an effective solution for dynamic traffic information acquisition and a reliable V2X communication network. Thirdly, successful implementation requires a holistic consideration and integration of hardware, software, and communication, as well as collaboration between cloud and edge computing. Finally, we also need to design effective methods for functionality verification and performance assessment, which involves substantiating functional capabilities through proof-of-concept (PoC) experiments and appraising the benefits brought by SMDT platform via large-scale traffic simulations.

As for the vehicle-road collaboration, we introduce the concept of local DT (LDT) on the basis of SMDT platform to some existing challenges. With RSU's fixed positions and continuous sensing capabilities, RSUs can monitor the driving behaviors within certain coverage areas. In this thesis, we also endow RSU with edge computing and V2X communications, so RSU serves as the infrastructure of LDT that can share perception information with surrounding vehicles, and learn and accumulate local human driving knowledge over time. In this setting, there is no real-time interaction between vehicle agents and the traffic environment. To achieve this goal, we propose an innovative RSU-based offline reinforcement learning (RL) framework to enable offline model training and online inference for hybrid vehicle control. Hardware-in-the-loop (HiL) simulations and PoC field trial results demonstrate that the proposed system offers a substantial enhancement in traffic safety and efficiency within RSU-installed areas, outperforming conventional autonomous driving systems.

The following parts of this thesis can be summarized as follows.

- Chapter 2: State-of-the-art Autonomous Driving and Mobility Digital Twins

    1. Surveys the state-of-the-art (SOAT) of autonomous driving, highlighting the importance of navigation, environmental perception, localization, prediction, planning, and motion control modules.
    2. Introduces general concepts and technologies about introducing DT into autonomous driving, then analyzes key challenges.

- Chapter 3: Smart Mobility Digital Twin: System Design and Implementation

    1. Develops a novel system architecture of the SMDT platform designed to address safety and robustness considerations. This architecture integrates cloud and edge computing across RSUs, CAVs, and a central cloud.
    2. proposes comprehensive system design and implementation. We detail the intricacies of core software and hardware components, as well as pivotal technologies for realizing SMDT platform in a real-world autonomous driving environment. Then we achieve the implementation in the test field, including a cloud server and distributed intelligent RSU and CAV edge with sensors, communication modules, and edge computing capabilities.
        - These works are published in

        * K. Wang, Z. Li, T. Yu, and K. Sakaguchi, "Smart Mobility Digital Twin for Automated Driving: Design and Proof-of-Concept," *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, Florence, Italy, pp. 1-6, 2023.

- Chapter 4: Global Digital Twin-Enabled Autonomous Vehicle Navigation Systems

    1. Introduces a CAV navigation system as one use case of the SMDT platform. By leveraging global information from SMDT, we design a cloud-based cooperative event-triggered route planning service for improving travel efficiency of CAVs.
    2. Demonstrates PoC on SMDT-enabled global route planning service to validate the system functionalities, also evaluates the communication reliability and latency based on standards for V2X use cases proposed by the 3GPP.
    3. Conducts large-scale traffic simulation to study the benefits of the proposed SMDT platform in terms of traffic efficiency and safety when using the proposed CAV navigation system.
        - These works are published in
            * K. Wang, T. Yu, Z. Li, K. Sakaguchi, O. Hashash, and W. Saad, "Digital Twins for Autonomous Driving: A Comprehensive Implementation and Demonstration," *2024 International Conference on Information Networking (ICOIN)*, Ho Chi Minh City, Vietnam, pp. 452-457, 2024.
            * K. Wang, Z. Li, K. Nonomura, T. Yu, K. Sakaguchi, O. Hashash, and W. Saad, "Smart Mobility Digital Twin Based Automated Vehicle Navigation System: A Proof of Concept," in *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 3, pp. 4348-4361, March 2024.

- Chapter 5: Local Digital Twin-Assisted Hybrid Autonomous Driving Systems

    1. Introduces a hybrid autonomous driving system as another use case of the SMDT platform. By using local DTs at RSU edge, RSUs can learn and accumulate local human driving knowledge over time, then facilitate intersection-specific local driving strategies for autonomous vehicles
    2. designs an offline RL framework based on V2I communications, including an environment perception data encoder and a kinematic-model-based decoder, is designed across RSU and CAV edge systems.

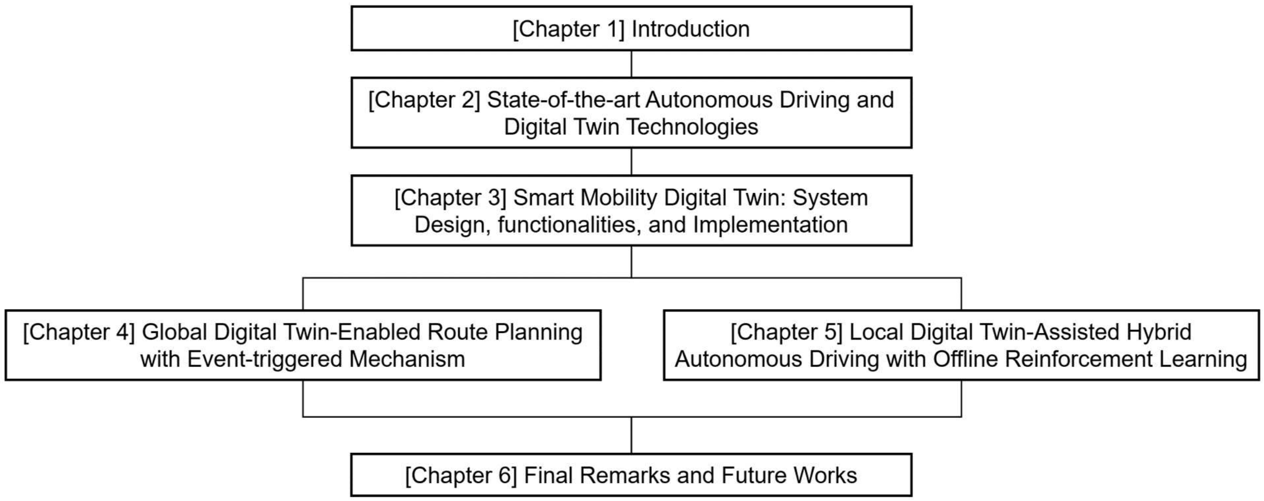


Figure 1.8: Thesis organization.

3. demonstrates the proposed system with significant improvements with safety enhancement of about 10% and travel time reduction of about 15%, compared to conventional autonomous driving systems. The implemented system also aligns with 3GPP requirements and demonstrates reliable real-time performance, fulfilling the targets of the proposed design.
    - These works are published in
        * K. Wang, C. She, Z. Li, T. Yu, Y. Li, and K. Sakaguchi, "Roadside Units Assisted Localized Automated Vehicle Maneuvering: An Offline Reinforcement Learning Approach," *27th IEEE International Conference on Intelligent Transportation Systems (IEEE ITSC 2024)*, Edmonton, Canada, 2024.
        * K. Wang, K. Nonomura, Z. Li, T. Yu, K. Sakaguchi, O. Hashash, W. Saad, C. She, and Y. Li, "Augmented Intelligence in Smart Intersections: Local Digital Twins-Assisted Hybrid Autonomous Driving," in *IEEE Transactions on Intelligent Vehicles*, Early Access Article.

Figure 1.8 shows the organization of this thesis.

Chapter 2

State-of-the-art Autonomous Driving and Mobility Digital Twins

This chapter establishes the required technical background and challenges of this thesis. First, in Section 2.1, we present the fundamental modules of an autonomous vehicle system with state-of-the-art (SOTA) technologies, i.e., navigation (route planning), environmental perception, localization, prediction, (local) planning, and motion control. Then, in Section 2.2, we introduce the main concepts and techniques in the establishment of mobility digital twins (DTs), including edge/cloud computing, vehicle-to-everything (V2X) communications, and some potential standards for the mobility DTs. In Section 2.3, we explain some technical challenges in the construction of smart mobility DT (SMDT). Finally, we summarize the chapter in Section 2.4.

2.1 SOTA Autonomous Driving

In this section, we describe the main modules in a modular autonomous vehicle system (ADS). An autonomous vehicle is composed of a number of subsystems, each responsible for a different task. An overview of a typical modular ADS architecture is depicted in Figure 2.1. We highlight six software modules (i.e., route planning, environmental perception, localization, prediction, planning, and motion control.) and two hardware modules (onboard sensors and vehicle actuator).

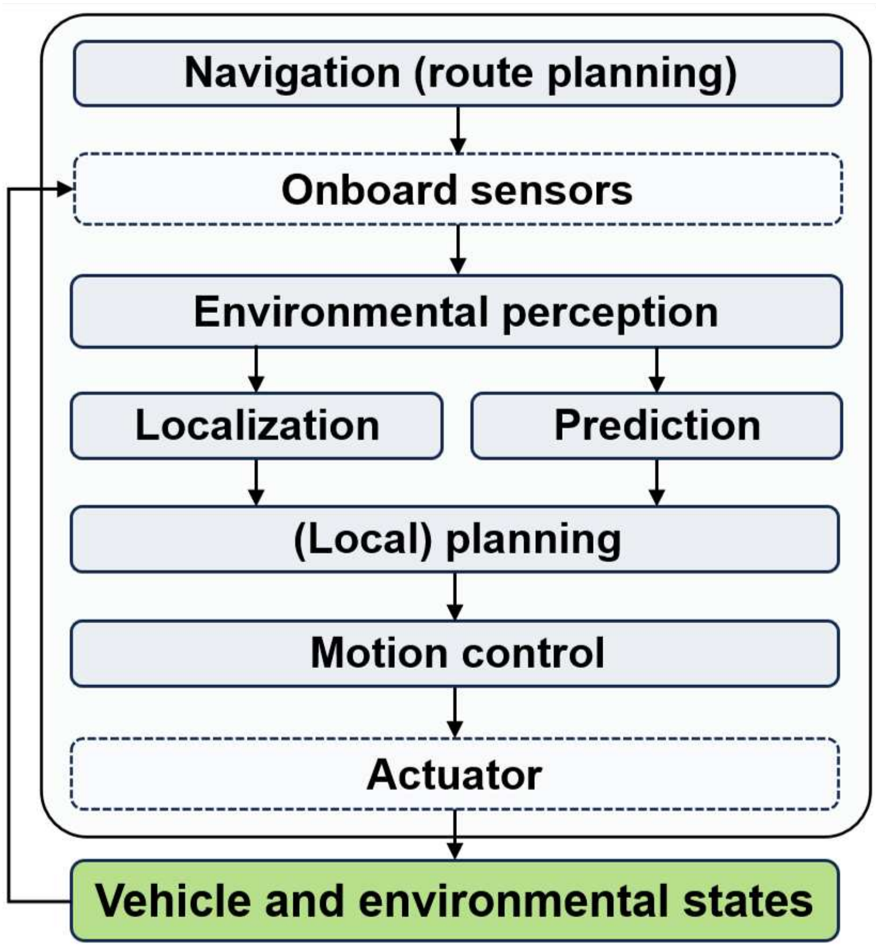
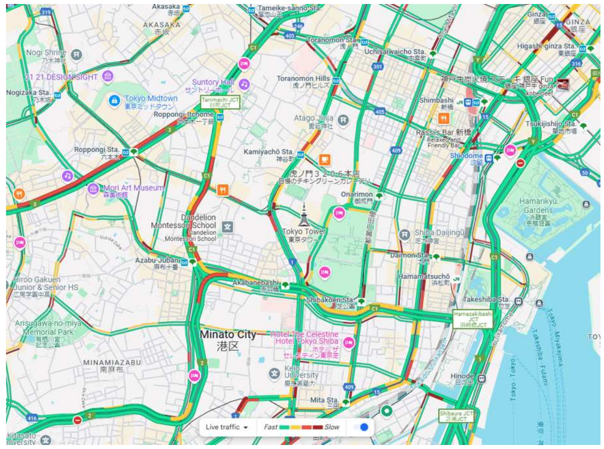


Figure 2.1: Overview of a modular ADS architecture.

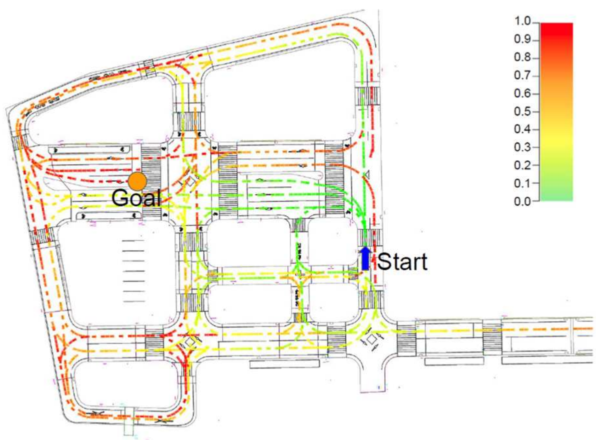
2.1.1 Navigation

An autonomous vehicle navigation system, i.e., route planner, produces high-level plans to achieve long-term goals, such as navigating between two locations with the fastest, shortest, or the most fuel-efficient route based on the road network. The planner also determines which roads to take, the appropriate lanes to stay in, and recommended speeds, similar to what commercially available GPS navigation services provide, e.g. Google Maps [83], Baidu Maps [84], and Mapbox [85].

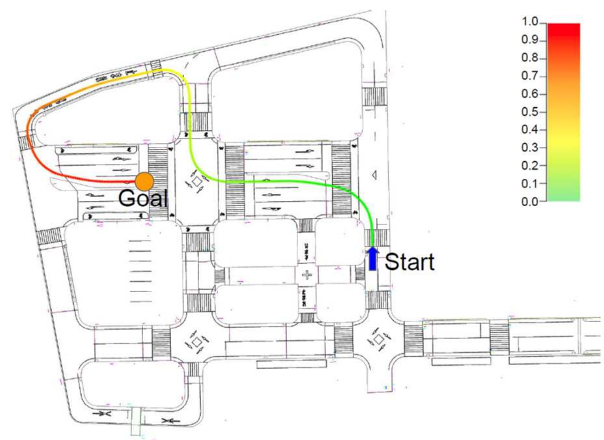
Route planning algorithms often represent the road network as a directed graph, where edges correspond to road segments, and weights represent the associated travel costs (e.g., time, distance, or traffic conditions). Classical approaches, such as Dijkstra's algorithm and A*, remain foundational but struggle with the computational demands of large-scale graphs,



(a) Live traffic service at Google Maps.



(b) Searching for the shortest path at OpenPlanner.



(c) The shortest path at OpenPlanner

Figure 2.2: A comparison of human-centered and ADS navigation systems: (a) Google Maps integrates real-time traffic data and calculates ETA for each road segment. (b) and (c) Searching the the shortest path at OpenPlanner, where line color indicates route distance, green color is the closest distance and red is the max to reach the goal position [82, 87].

which may contain millions of edges. Modern navigation systems enhance these methods by integrating real-time traffic data, statistical models, and predictive analytics to predict estimated time of arrival (ETA), ensuring accurate and efficient route suggestions. For instance, Google Maps and Baidu Maps use data-driven approaches to analyze historical traffic patterns and provide dynamic updates based on current conditions [86,87].

In the context of autonomous driving, the navigation system has also made significant advancements in recent years. Modern frameworks like OpenPlanner aim to provide the shortest path based on the static high-definition (HD) maps [88]. However, unlike human-centered navigation systems, which integrate real-time traffic conditions and historical data to optimize routes dynamically, autonomous vehicle frameworks generally lack the ability to incorporate such dynamic traffic information into their route planning. This limitation arises due to the challenge of integrating real-time sensor data, V2X communication, and centralized traffic updates with onboard decision-making processes. As depicted in Figure 2.2, we show the live traffic service in Google Maps and how OpenPlanner searches the shortest route.

As a result, autonomous driving systems predominantly rely on static maps for route planning and are unable to proactively adjust to evolving traffic conditions, such as sudden congestion, accidents, or temporary road closures. Bridging this gap between static map-based planning and dynamic traffic-aware navigation remains a critical challenge in the development of truly adaptive and efficient ADS.

2.1.2 Onboard Sensors

Autonomous vehicles rely on an array of onboard sensors, which can be categorized into exteroceptive sensors for perceiving surroundings and proprioceptive sensors for internal vehicle state monitoring. These sensors enable the vehicle to make real-time decisions by providing rich data about the environment, vehicle dynamics, and internal states. We summarize the onboard sensors most commonly used in autonomous driving.

- **Camera:**

    1. *Monocular Camera* is a passive sensor that captures color images and is widely used for traffic light recognition, lane detection, and general object recognition. However, their reliance on illumination brings them poor performances in darkness or adverse weather conditions. Additionally, though some studies [89] use deep convolutional neural networks (CNN) to improve monocular camera-based depth perception, it is still difficult to obtain accurate depth information with a single monocular camera

    2. *Depth Camera*, such as stereo cameras or RGB-D cameras, provide both color and depth information. These sensors are particularly useful for estimating the 3D structure of the environment, enabling tasks like obstacle detection, distance measurement, and scene reconstruction.

3. *Event Camera* is emerging as an innovative solution in ADS. Unlike traditional frame-based cameras, event cameras capture changes in pixel intensity asynchronously. They operate with high temporal resolution, low latency, and minimal power consumption, making them ideal for dynamic environments and high-speed scenarios. However, their limited resolution and large pixel sizes constrain their broader adoption. Efforts to integrate event camera data into driving datasets have begun to address these challenges [90].

- **Radar (Radio Detection and Ranging):**

  1. *Millimeter-Wave (mmWave) Radar* operates in the 77-81 GHz and is compact enough to fit inside vehicle components like side mirrors. It is a highly reliable active sensor that emits radio waves and measures their reflections to detect objects and calculate their distance. It is cost-effective, lightweight, and capable of long-range detection. While radar can work well in adverse weather conditions, it lacks the precision of LiDAR for detailed mapping. Hence, Radars remain an essential complement to cameras, filling gaps in depth perception and illumination-independent sensing.
  2. *Imaging Radar* provides higher resolution and better angular accuracy, enabling detailed object recognition and classification in addition to traditional range and speed measurements [91].

- **LiDAR (Light Detection and Ranging)** is a cornerstone technology for ADS, offering high-resolution 3D spatial data by emitting infrared light waves. It is particularly effective for short-to-medium-range sensing under 200 meters, where its accuracy surpasses radar. However, the performance of LiDAR is negatively affected by extreme weather conditions such as fog and snow. Although LiDAR systems are larger and more expensive than camera or Radar, with the anticipated widespread adoption of autonomous driving, it can be expected that the cost of LiDAR will significantly decrease with a smaller design, becoming as reasonable as the cost of cameras today [92]. As shown in Table 2.1, we compare some existing exteroceptive sensors in detail.

- **GPS (Global Positioning System):**

  1. *Real-time kinematic (RTK) GPS* enhances standard GPS accuracy by using cor-

rections from ground-based reference stations, which achieves centimeter-level precision, critical for lane-level navigation.

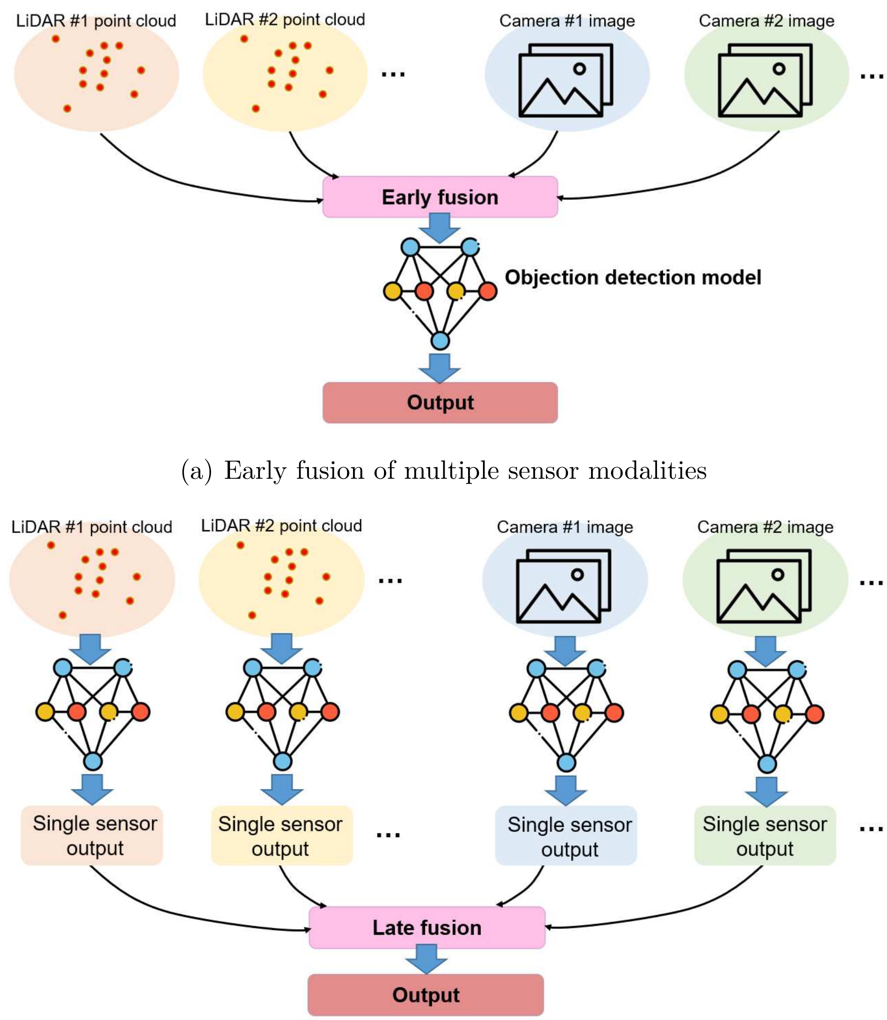
2. *Global navigation satellite system (GNSS)* integrates multiple satellite systems (e.g., GPS, GLONASS, Galileo) to improve positioning reliability, especially in urban canyons or dense forested areas.

- **Proprioceptive Sensors** monitor the vehicle's internal states, such as speed, acceleration, yaw, and altitude. Proprioceptive data is often accessed through the controller area network (CAN) bus protocol in modern vehicles.

  1. *Wheel encoders* are used for odometry, aiming at calculating the traveled distance.
  2. *Inertial measurement units (IMUs)* provide data on longitudinal acceleration, lateral acceleration, angular velocity, and orientation, essential for navigation and stabilization.
  3. *Tachometers and altimeters* measure speed and altitude, respectively, offering additional feedback for vehicle control.

Table 2.1: Comparison of existing onboard exteroceptive sensors

Sensors	Camera			Radar		LiDAR
	Monocular camera	Depth camera	Event camera	Mmwave radar	Image radar	
Color	✓	✓	✗	✗	✗	✗
Depth	✗	✓	✗	✓	✓	✓
Accuracy	Low	Low	High (time) Low (space)	Middle	High	High
Range	Low (<100m)	Low (<100m)	Low (<100m)	High (<300m)	Medium (<200 m)	Medium (<200 m)
Size	Very small	Medium	Very small	Small	Small	Large
Cost	Lowest	Low	High	Middle	Middle	High
Illum. affection	✓	✓	✗	✗	✗	✗
Weather affection	✓	✓	✓	✗	✗	✓



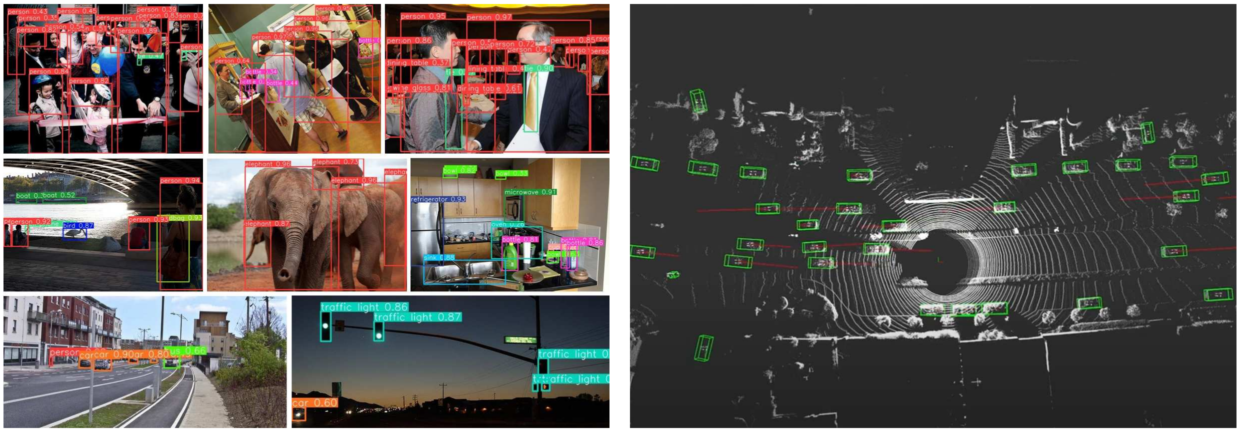
(a) Early fusion of multiple sensor modalities

(b) Late fusion of multiple sensor modalities

Figure 2.3: Two paradigms for onboard sensor fusion

2.1.3 Sensor Fusion and Environmental Perception

The environmental perception module is responsible for classifying, detecting, and tracking surrounding objects based on multiple sensory data. Hence, this module should also fuse sensory data into a consistent and suitable representation of the world for autonomous driving. As shown in Figure 2.3, sensor fusion can be broadly categorized into early fusion and late fusion. Early fusion involves integrating raw data from sensors, such as point clouds from LiDAR and images from cameras, at the data acquisition stage. This approach preserves the richness of sensor data, enabling more detailed and nuanced interpretations. Late fusion,



(a) Camera-based object detection with YOLOv10 (b) LiDAR-based object detection with CenterPoint

Figure 2.4: Examples of onboard sensor-based object detection [94,95]

on the other hand, combines the outputs of individual sensor-specific models or algorithms. This method reduces the computational burden and simplifies implementation by allowing each sensor to be processed independently before their results are merged. In addition, in cooperative perception scenarios, by transmitting only processed intermediate results, such as object-level detections or feature vectors rather than raw sensor data, late fusion can significantly save communication resources between vehicles or smart infrastructures. Both strategies play complementary roles in achieving reliable environmental perception, with early fusion excelling in scenarios requiring precise spatial understanding and late fusion being more practical in modular and cooperative system designs.

Advanced perception algorithms have witnessed significant advancements, with state-of-the-art methods tailored to specific sensor modalities. For LiDAR-based object detection, CenterPoint [93] has emerged as a leading approach. This algorithm models 3D objects as points and uses a keypoint detector to identify object centers, followed by regression to predict attributes such as size, orientation, and velocity. CenterPoint has demonstrated superior performance on benchmarks like nuScenes and Waymo, showcasing its robustness and accuracy in complex environments. In the realm of camera-based object detection, the YOLO (You Only Look Once) [94] series is one of the most influential algorithms. YOLO treats object detection as a single regression problem, directly predicting bounding boxes and class probabilities from full images in one evaluation. This unified architecture enables real-time processing, with the original YOLO model processing images at 45 frames per second. The latest version, YOLOv10 [95], introduces innovations such as non-maximum suppression (NMS)-free training and lightweight architectures, offering faster inference and higher accuracy compared to its predecessors, which makes YOLOv10 particularly well-suited for applications requiring low-latency data processing like autonomous driving. As depicted in Figure 2.4, we show some examples of camera and LiDAR-based object detection using CenterPoint and YOLOv10 [95,96].

Object tracking, also referred to as multiple object tracking (MOT), is another important field in environmental perception, For ADS in complex or high-speed scenarios, estimating surrounding objects' location alone is insufficient. It is necessary to estimate their heading angle and velocity so that a motion model can be applied to track the object over time and predict future trajectory to avoid collisions. Traditional tracking methods often employ data association techniques combined with filtering algorithms [97]. Recent advancements have led to the development of more sophisticated tracking algorithms that leverage deep learning and sensor fusion. For example, the CenterRadarNet [98] framework integrates 4D frequency-modulated continuous-wave (FMCW) radar data with deep learning models to perform joint 3D object detection and tracking. Another notable method is the TripletTrack [99] algorithm, which employs triplet embeddings and long short-term memory (LSTM) networks for 3D object tracking. By learning discriminative features through triplet loss and modeling temporal dependencies with LSTMs, TripletTrack effectively re-identifies objects and maintains robust tracking, even in challenging scenarios involving occlusions and reappearances.

Despite these innovations, environmental perception systems still face considerable challenges. Adverse weather conditions, including rain, fog, and snow, can degrade the reliability of sensors like LiDAR and cameras. While radar provides robust performance in such conditions, integrating data across varying sensor reliabilities remains a complex task. Additionally, the computational demands of processing high-resolution sensor data in real-time pose significant challenges, particularly in urban environments with dense traffic and dynamic obstacles. Emerging approaches leveraging large vision models (LVMs) [100], such as pre-trained transformers, offer promising solutions by improving robustness and generalization in perception systems.

2.1.4 Localization

An accurate vehicle localization is crucial for precise, safe, and comfortable vehicle motion, enabling vehicles to determine their precise state (e.g., position, orientation, and speed) within

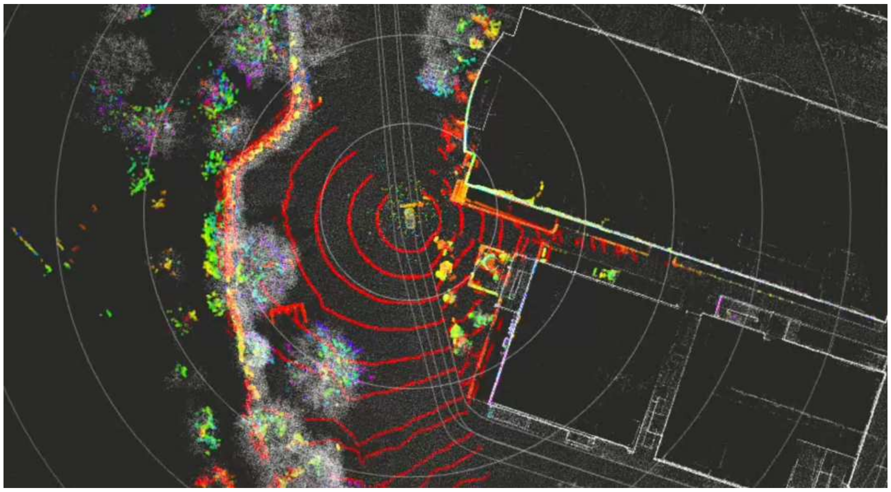


Figure 2.5: An example of LiDAR-based localization using NDT scan matching.

an environment. Localization methods can be broadly divided into map-based localization and mapless localization. Each approach has its own advantages and limitations, and hybrid methods that combine both are also being explored.

HD map-based localization utilizes detailed, high-resolution maps that contain comprehensive information about the environment, including road geometry, lane markings, traffic signs, and other static features. Vehicles equipped with sensors such as LiDAR or radar collect real-time data, which is then matched against the static HD map to determine the vehicle's position with high accuracy (see [101,102]). This process often involves algorithms like simultaneous localization and mapping (SLAM) and normal distributions transform (NDT) scan matching to continuously update the vehicle's location. For instance, NDT-LOAM [103], a LiDAR-based NDT matching approach, can achieve centimeter-level accuracy with a frequency of 10 Hz in real time. As shown in Figure 2.5, real-time point cloud generated by LiDAR can be accurately matched with a point cloud map to achieve vehicle localization. However, the reliance on HD maps presents challenges, including the need for map generation, substantial storage capacity, and the necessity for frequent updates to reflect changes in the environment. Hence, methods such as vehicle-to-infrastructure (V2I)-based map distribution have been proposed, where real-time HD maps are transmitted to vehicles from smart infrastructure, reducing the need for onboard storage [104].

Mapless localization does not depend on pre-existing maps, but instead relies on real-time sensor data to estimate the vehicle's state. Commonly used sensors include GPS, IMU, and wheel encoders. Data from these sensors are fused using algorithms such as the Kalman Filter to provide continuous position estimates. For example, GPS provides absolute positioning data, while IMUs offer information on acceleration and rotation, and wheel encoders measure wheel rotation to estimate traveled distance. The Kalman Filter integrates these inputs to produce a more accurate and reliable position estimate than any single sensor could achieve alone. This approach is particularly useful in environments where HD maps are unavailable or impractical to maintain. The biggest challenge in mapless methods exists in the reliability of GPS, since GPS usually has low accuracy near tall buildings or does not work well underground.

Recognizing the limitations inherent in both HD map-based and mapless localization, hybrid approaches have been developed to leverage the strengths of each, by utilizing HD maps when available to provide high-precision localization and revert to mapless methods in areas where static maps are outdated or unavailable. Advancements in sensor fusion techniques have enabled the integration of data from multiple sensors, such as cameras, LiDAR, radar, GPS, and IMUs, to enhance localization accuracy and robustness [105,106]. This multi-sensor fusion approach also allows the system to compensate for the weaknesses of individual sensors, such as GPS signal loss in urban canyons or LiDAR performance degradation in adverse weather conditions.

2.1.5 Prediction

In the context of autonomous driving, trajectory prediction enables vehicles to anticipate the future behaviors of surrounding agents such as vehicles, pedestrians, and cyclists. Accurate trajectory prediction is essential for safe and efficient local planning and vehicle control.

Early approaches to trajectory prediction primarily relied on kinematic and statistical models. Kinematic models assume constant velocity or acceleration to forecast future positions, which are computationally efficient but lack the flexibility to account for sudden changes in vehicle behaviors or complex interactions between agents. Statistical approaches, such as Gaussian processes and hidden Markov models, have been employed to model uncertainties and learn patterns from historical data, but fail to generalize well to diverse driving scenarios.

Recent advancements in deep learning have introduced Transformer architectures [107] as a powerful tool for trajectory prediction. Transformers, originally developed for natural language processing, excel at capturing long-range dependencies and handling sequential data, making them well-suited for modeling the temporal and spatial dynamics inherent in trajectory prediction. For instance, the Lane Transformer model [107] employs a transformer

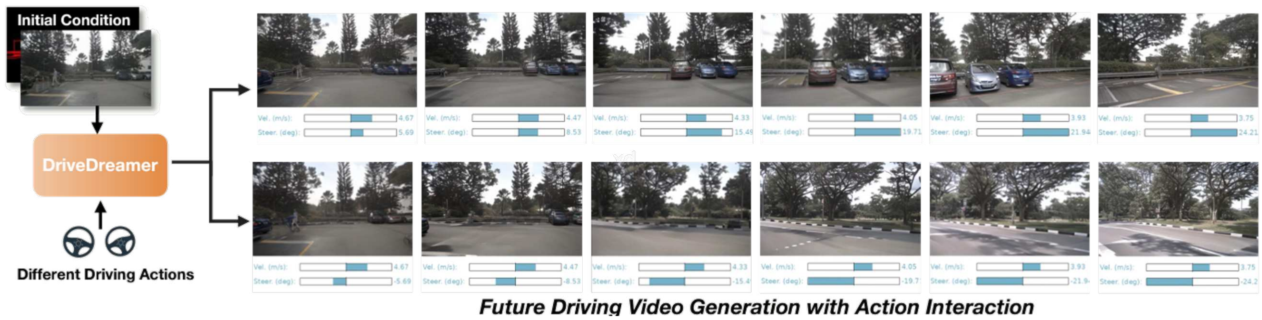


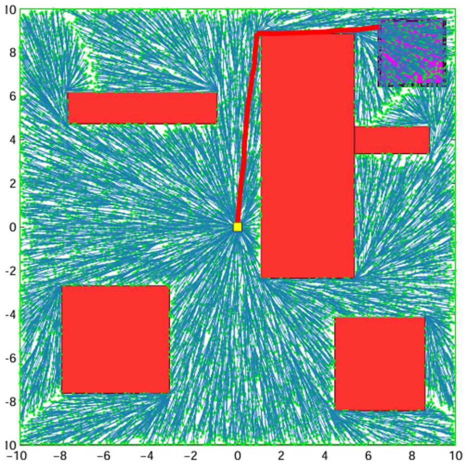
Figure 2.6: Future driving video generation with action interaction in DriveDreamer [110].

architecture to efficiently predict vehicle trajectories by modeling interactions between agents and their surrounding lanes. This approach has demonstrated high performance on public datasets, achieving both accuracy and computational efficiency. Similarly, MacFormer [108] incorporates map constraints into trajectory prediction, using novel modules, such as a coupled map and reference extractors, along with a multi-task optimization strategy (MTOS), to model topology and rule constraints effectively. This method also achieves SOTA results on benchmarks like Argoverse and nuScenes, offering robust and real-time prediction capabilities with a lightweight architecture. In addition, graph neural networks (GNNs) can model interactions between multiple agents by representing them as nodes in a graph with edges capturing their relationships, which is particularly useful in multi-agent scenarios where understanding the behavior of surrounding vehicles is crucial [109].

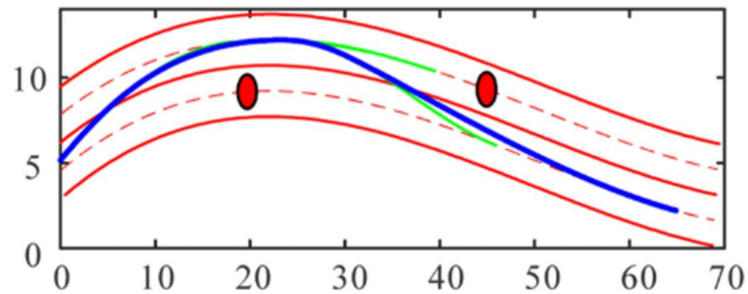
In recent years, the emergence of world models has further revolutionized trajectory prediction by enabling simulation-driven forecasting [110]. A prime example is DriveDreamer [111], which builds a world model entirely from real-world driving scenarios. DriveDreamer employs an autonomous-driving diffusion model (Auto-DM) to generate high-quality driving video predictions, demonstrating the ability to anticipate both dynamic object trajectories and environmental variations, thereby offering precise and controllable predictions that are closely aligned with real-world traffic conditions, as shown in Figure 2.6.

2.1.6 Local Planning

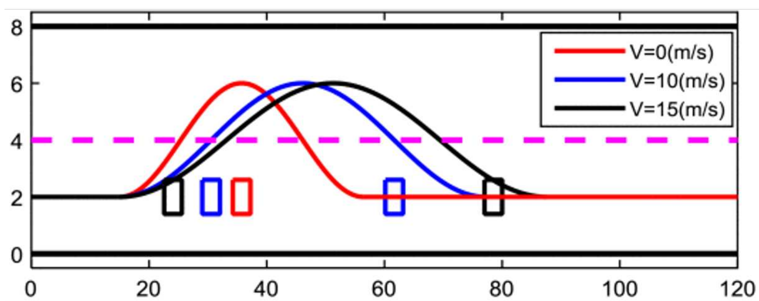
Planning is responsible for generating safe, efficient, comfortable, and dynamically feasible paths or trajectories for vehicles, where a path planner only includes kinematic consideration, but a trajectory (motion planner) is also time-parameterized. Considering that autonomous driving is inherently a temporal process, here we focus on *motion planning* techniques.



(a) RRT* tree with several obstacles



(b) Bézier curve on double-lane road with two static obstacles



(c) MPC-based trajectories with different obstacle speeds

Figure 2.7: Examples of sampling-based, interpolation-based, and optimization-based trajectory planning [111,115,117].

In general, traditional motion planning has been addressed through sampling, interpolation, and optimization-based approaches. More recently, artificial intelligence (AI)/machine learning (ML) methods have also been widely applied.

Sampling-based methods construct a representation of the feasible motion space by randomly sampling possible states or control inputs, particularly effective in high-dimensional spaces where explicit enumeration of all possible states is computationally infeasible. For instance, rapidly-exploring random trees (RRT) and its variant RRT* [112] incrementally build a search tree by randomly sampling the state space and extending branches toward these samples while avoiding obstacles, which is efficient for finding feasible paths in complex environments. Recent advancements have focused on improving the efficiency and applicability of sampling-based methods. Techniques such as informed RRT* [113] and batch informed trees

(BIT) [114] incorporate heuristics to guide the sampling process toward regions that are more likely to contain optimal paths, thus reducing computation time.

Interpolation-based methods generate smooth trajectories by connecting waypoints using mathematical functions that ensure continuity and differentiability, essential for vehicle control and passenger comfort. For example, spline-based planning, including the use of cubic and quintic polynomials, is widely used to interpolate between waypoints, providing smooth paths that satisfy position, velocity, and acceleration constraints [115]. Bezier curves and B-splines are also employed to generate flexible and smooth trajectories [116]. Clothoid curves, or Euler spirals, are curves with linearly changing curvature, making them suitable for modeling paths that vehicles can follow with continuous steering inputs. They are especially useful in highway and lane-changing scenarios where smooth curvature transitions are required [117]. While interpolation-based methods are computationally efficient and produce high-quality paths, they may struggle in environments with complex obstacle configurations or dynamic obstacles, as they do not inherently account for collision avoidance during interpolation.

Optimization-based planners formulate motion planning as an optimization problem, seeking to minimize a cost function while satisfying a set of nonholonomic constraints representing vehicle dynamics and environmental factors. Model predictive control (MPC) predicts future vehicle states over a finite horizon and solves an optimization problem at each time step to find the control inputs that minimize a cost function, such as deviation from the desired path while satisfying constraints like vehicle dynamics and obstacle avoidance [118,119]. Convex optimization techniques are also leveraged to formulate the planning problem, allowing efficient solvers to find globally optimal solutions in real time. Quadratic programming (QP) and second-order cone programming (SOCP) are commonly used, especially when the cost function and constraints can be linearized or approximated to maintain convexity [120,121].

AI/ML methods introduce data-driven approaches that can enhance the adaptability and robustness of planning. For example, imitation learning like behavior cloning (BC) trains the model to mimic the behavior of expert drivers [122]. While straightforward, imitation learning can suffer from distributional shifts if the vehicle encounters states not represented in the training data. Algorithms like deep Q-networks (DQN) and proximal policy optimization (PPO) have also been applied to motion planning tasks [123,124]. Hybrid approaches combine traditional planning methods with AI/ML techniques to leverage the strengths of both. For instance, ML models can be used to predict the intentions of other agents or estimate the cost-to-go, which can be integrated into optimization-based planners to improve performance in dynamic environments [125]. While AI/ML-based methods offer the potential to handle complex, unstructured environments and learn from vast amounts of data, challenges remain in ensuring safety, interpretability, and generalization. Robust validation and verification processes are essential to deploy these methods in safety-critical scenarios.

2.1.7 Motion Control and Vehicle Actuator

The motion control is crucial in the design of ADS since it is responsible for stabilizing the vehicle and executing planned paths or trajectories by generating appropriate control commands for the vehicle actuators. One of the first proposed solutions to deal with path following problem is pure-pursuit control [126], which is a geometric path-tracking algorithm designed to minimize the lateral error between the vehicle's position and a look-ahead point on the desired path, thereby ensuring smooth and accurate path following. The pure-pursuit controller is favored for its simplicity, computational efficiency, and ease of implementation, hence, it has been widely adopted in autonomous driving research and applications, such as two participant teams in the Defense Advanced Research Projects Agency (DARPA) Grand Challenge [127] and three in DARPA Urban Challenge [128]. However, the negligence of vehicle dynamics and lack of obstacle avoidance capability have greatly limited the performance of pure-pursuit controllers. Other traditional methods include proportional-integral-derivative (PID) and linear quadratic regulators (LQR) control. While traditional methods provide a solid foundation, they may struggle with the nonlinearities, constraints, and complexities of real-world autonomous driving scenarios, which has led to the development of more advanced control techniques like MPC.

MPC optimizes control inputs over a finite prediction horizon by solving an optimization problem at each time step, accounting for system dynamics, constraints, and future reference trajectories, making it well-suited for motion control in autonomous vehicles. At each time step t, MPC solves the following optimization problem:

$$
\begin{aligned}
\min_{\boldsymbol{u}_{t:t+N-1}} \quad & \sum_{k=0}^{N-1} \boldsymbol{s}_{t+k|t}^{\top} \mathbf{Q} \boldsymbol{s}_{t+k|t} + \boldsymbol{u}_{t+k}^{\top} \mathbf{R} \boldsymbol{u}_{t+k} \\
\text{s.t.} \quad & \mathbf{s}_{t+k+1|t} = f(\boldsymbol{s}_{t+k|t},\ \boldsymbol{u}_{t+k}), \quad k = 0, \dots, N-1 \\
& \boldsymbol{x}_{t|t} = \boldsymbol{x}(t) \\
& \boldsymbol{s}_{t+k|t} \in \mathcal{S}, \quad \boldsymbol{u}_{t+k} \in \mathcal{U}, \quad k = 0, \dots, N-1
\end{aligned}
\tag{2.1}
$$

where $\boldsymbol{s}_{t+k|t}$ is the predicted state at time $t+k$ based on state at time t. $\boldsymbol{u}_{t+k}$ is the control input at time $t+k$. N is the prediction horizon. $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices for states and control inputs. $\mathcal{S}$ and $\mathcal{U}$ are convex polytopes and represent state and control constraints, respectively.

If the system dynamics are linear (i.e., $f(\cdot)$ is a linear function that can be represented as $\boldsymbol{s}_{t+k+1|t} = \mathbf{A}\boldsymbol{s}_{t+k|t} + \mathbf{B}\boldsymbol{u}_{t+k}$) and the cost function is quadratic, the MPC optimization problem becomes a QP problem. In this case, efficient QP solvers can find the optimal control sequence quickly, making real-time implementation feasible.

When $f(\cdot)$ is nonlinear, the MPC optimization becomes a nonlinear programming (NLP) problem, which is more computationally intensive to solve. The computational complexity increases significantly with the prediction horizon N, which may not meet the real-time constraints in autonomous driving. In the most classic MPC-related research [129], the computational complexity and performance of the linear and non-linear schemes are discussed in detail.

Reinforcement learning (RL) presents a promising alternative by learning control policies from interactions with the environment, potentially handling complex and nonlinear dynamics without explicit models. The RL problem is often modeled as a Markov decision process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ represent the spaces of states and available actions. $\mathcal{T}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ defines the probability of transitioning to state $\boldsymbol{s}_{t+1}$ from state $\boldsymbol{s}$ after taking action $\boldsymbol{a}_t$ that describes the dynamics of the system. $\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is the reward function and $\gamma \in (0, 1]$ is a scalar discount factor. The objective of RL is to find a policy $\pi(\boldsymbol{a}|\boldsymbol{s})$ that maximizes the expected cumulative reward:

$$J(\pi) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \ \mathcal{R}(\boldsymbol{s}_{t+k}, \boldsymbol{a}_{t+k}) \ \middle| \ \pi\right] \tag{2.2}$$

Considering that RL effectively can handle high-dimensional state spaces in complex autonomous driving environments, overcoming limitations of traditional motion control methods, the application of RL and its variants in vehicle control field is increasingly being explored as a promising approach for achieving adaptive, efficient, and robust decision-making, such as deep RL (DRL) [130], offline RL [131], and inverse RL [132]. However, challenges like the curse of rarity (CoR), ensuring safety during exploration, and achieving robust generalization across diverse driving scenarios remain significant obstacles that require further research and innovative solutions to fully realize the potential of RL in real-world autonomous driving systems.

After the motion control module computes the desired control actions (e.g., $\boldsymbol{u}_t$ in (2.1) and

$\boldsymbol{a}_t$ in (2.2)), these commands are transmitted to the vehicle's actuators for execution. The control signals, typically representing steering angles, throttle positions, and braking forces, are interfaced with the vehicle's electronic control units (ECUs). The ECUs translate these high-level commands into low-level electrical signals that drive the actuators, such as electric motors for steering, fuel injection systems for acceleration, and hydraulic systems for braking. To maintain precision and compensate for external disturbances or model inaccuracies, proprioceptive sensors continuously monitor the vehicle's state, such as wheel speeds, yaw rates, and lateral acceleration, providing real-time feedback to the control system and enabling the vehicle to follow the control signals accurately.

2.2 Establishment of Mobility Digital Twin

In this section, we introduce the main concepts and techniques in the establishment of mobility DTs to integrate ADS and DT technologies. The concepts of edge/cloud computing, V2X communications, as well as communication standards of mobility DTs, are widely used in the technical chapters of this thesis.

2.2.1 Cloud/Edge Computing

The establishment of mobility DT relies heavily on advanced computational infrastructures capable of processing vast amounts of data generated by autonomous vehicles and the surrounding environment. Cloud and edge computing paradigms are instrumental in enhancing the performance of various modules in ADS, including navigation, sensing, environmental perception, localization, prediction, planning, and motion control. It is necessary to discuss how cloud and edge computing augment these modules, facilitate vehicle-road collaboration, and provide augmented intelligence to autonomous vehicles within the framework of mobility DT. The integration of cloud/edge computing in mobility DT along with its functionalities are shown in Figure 2.8.

2.2.1.1 Cloud Computing in mobility DT

Cloud computing offers scalable and centralized computational resources, making it ideal for handling large-scale data processing and complex computations required in autonomous driving. In recent years, the emergence of commercial cloud computing services, such as Amazon

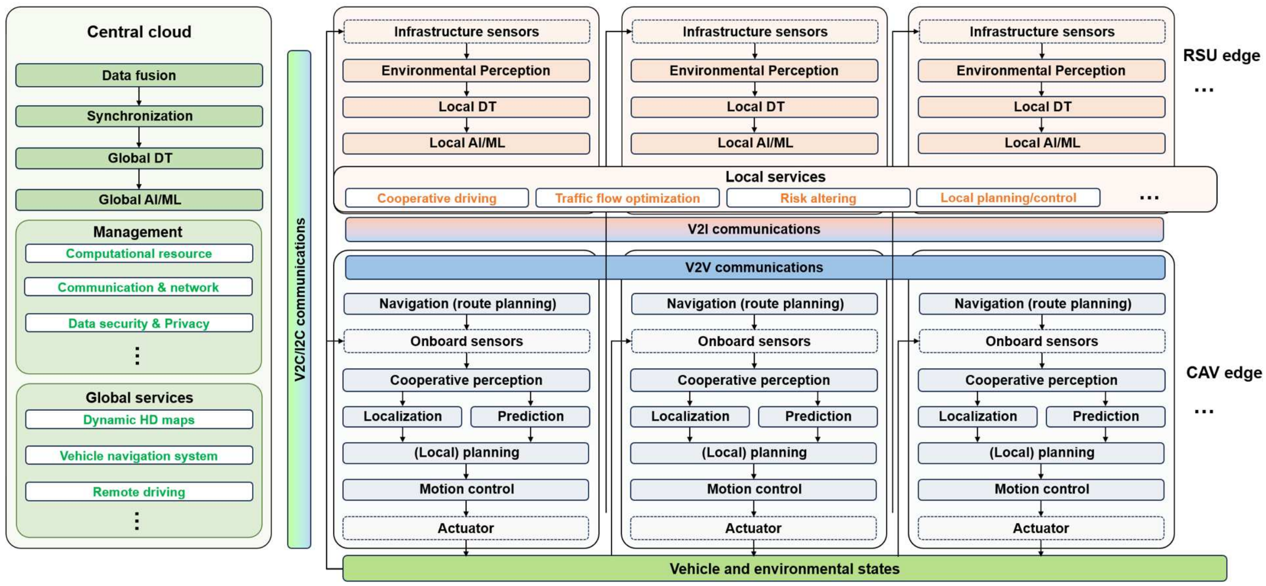


Figure 2.8: How cloud/edge computing in mobility DT enhances ADS

Web Services (AWS) [133], Google Cloud Platform (GCP) [134], and specialized platforms like Baidu's Apollo Cloud [135] and Tier IV's Web Auto [136], has significantly advanced applications within vehicular and transportation cyber-physical systems (CPS). These services provide a wide array of foundational infrastructure and tools for distributed computing, including computing power, storage solutions, networking capabilities, databases, analytics, and Internet of Things (IoT) integrations. The scalability and versatility of these cloud platforms enable connected vehicles to offload data processing and computational demands, facilitating more efficient and intelligent transportation systems. In the context of mobility DT, integrating the cloud is crucial for enhancing the efficiency of autonomous driving. Below, we discuss how cloud computing can be effectively incorporated into mobility DT to enhance autonomous driving through dynamic map updates, cooperative sensing and perception, and augmented intelligence via data fusion.

Data fusion and synchronization: The cloud serves as a central hub for aggregating and synchronizing data from diverse sources, including vehicles and infrastructure systems. Implementing data fusion algorithms in the cloud enables the combination of heterogeneous data types, such as LiDAR, radar, camera, and GPS information, to create a comprehensive and coherent representation of the environment. This centralized fusion facilitates the development of high-fidelity object-level digital twins that mirror real-world conditions. To achieve efficient data fusion and synchronization, timestamp alignment and spatial-temporal correlation methods should be employed. Cloud platform can utilize distributed databases and parallel processing techniques to handle the high data throughput and ensure that the fused data is consistent and up-to-date.

Resource management: Effective resource management in the cloud is crucial for allocating computational resources to various autonomous driving modules as needed, which involves implementing virtualization and containerization technologies, such as virtual machines (VMs) and Docker containers, to provide scalable and flexible computational environments. Resource scheduling algorithms prioritize tasks based on their computational demands and latency requirements, ensuring that critical modules receive the necessary resources promptly. Furthermore, cloud orchestration tools, like Kubernetes, manage containerized applications, enabling automated deployment, scaling, and operation of applications across different devices, facilitating the efficient management of resources for modules like environmental perception and prediction, which require substantial computational power. On the other hand, Managing network resources is essential for ensuring reliable and efficient data transmission between edge nodes and the cloud. Techniques such as software-defined networking (SDN) and Network Function Virtualization (NFV) should also be employed to dynamically allocate network bandwidth and optimize data flow. Quality of service (QoS) protocols prioritize network traffic for latency-sensitive applications, ensuring that critical data, such as control commands and safety-critical messages, are transmitted with minimal delay.

Cloud-based services: Integrating cloud computing into the mobility DT enables the development of various cloud-based services that enhance autonomous driving:

- *Dynamic HD maps:* The cloud can generate and update HD maps in real-time by processing data collected from multiple vehicles and infrastructure sensors, providing detailed information about road geometry, traffic signs, and dynamic traffic information, which are essential for vehicle navigation and planning modules.

- *Vehicle navigation systems:* Cloud-based navigation services utilize real-time traffic data and predictive analytics to provide optimal routing solutions. By integrating with the mobility DT, these services can adapt to changing conditions and provide vehicles with the most efficient routes.

- *Remote Driving:* In scenarios where autonomous vehicles encounter complex situations beyond their capabilities, remote driving services allow human operators to take control via cloud-based interfaces.

To implement these services effectively, standardized APIs and communication protocols are necessary to ensure interoperability between different system components.

2.2.1.2 Edge Computing in mobility DT

Edge computing brings computation and data storage closer to data sources, which is critical for real-time processing and safety-critical requirements in autonomous driving. Also, recent advancements in commercial edge computing devices, such as NVIDIA Drive [137] and Jetson [138] platforms, Lenovo ThinkSystem SE350 [139], Intel's NUC [140], and Raspberry Pi [141], offer several advantages like compact size, high computational power, and energy efficiency, making them ideal for integration into mobility DT. Integrating these edge computing devices into mobility DT involves deploying computational resources on vehicles and infrastructure to handle immediate data processing needs, supporting real-time tasks such as environmental perception, localization, planning, and control. By bringing computation closer to the data sources, edge computing reduces reliance on cloud connectivity and enhances the responsiveness and reliability of autonomous driving systems. Below, we discuss how edge computing can be incorporated into mobility DT to enhance data-plane sensing and perception, cooperative localization, and vehicle-road collaboration for planning and control.

Local DT for cooperative sensing and perception: At the edge, local DTs can be created to model immediate environments around infrastructure locations, by processing sensor data in real-time to provide precise and up-to-date representations of the surroundings, which are essential for cooperative sensing and perception. Implementing edge-based data fusion techniques allows vehicles and RSUs to share and integrate sensor data, enhancing the detection of obstacles and improving situational awareness. Protocols like cooperative perception messaging (CPM) can be utilized to facilitate the exchange of perception data between entities at the edge.

Local AI/ML-based local services: Edge computing enables the deployment of AI/ML models directly on vehicles and RSUs to provide immediate services:

- *Cooperative driving:* Edge devices execute algorithms for cooperative maneuvers, such as platooning and coordinated lane changes, by processing local data and communicating with nearby vehicles.

- *Traffic flow optimization:* Edge nodes analyze local traffic conditions and adjust signal timings or provide routing suggestions to improve traffic efficiency.

- *Risk alerting:* Real-time analysis of sensor data at the edge allows for immediate detection of hazardous conditions, such as sudden stops or blind spots, enabling prompt alerts to nearby vehicles.

- *Local planning and control:* Edge computing supports extracting and learning human-driven vehicles' behaviors, thereby enabling the generation of local driving strategies in the format of CNN models. Then such models can be transmitted to CAVs for local maneuvering.

To implement these services, lightweight AI/ML models optimized for edge devices should be employed, and hardware acceleration techniques, such as using graphics processing units (GPUs), should also be utilized to enhance processing capabilities.

2.2.2 V2X Communications

In the mobility DT, V2X communications are crucial for enabling seamless interaction among vehicles, infrastructure, and central cloud. The primary V2X modalities in mobility DT include *V2I, vehicle-to-network (V2N), and infrastructure-to-network (I2N)* communications. It is necessary to analyze each modality, exploring how they integrate within the mobility DT to support autonomous driving modules and services.

V2I communications involve the exchange of information between vehicles and roadside infrastructure, such as traffic signals, road signs, and RSUs. In the context of mobility DT, V2I communications support a variety of services, each with distinct data requirements and quality of service (QoS) constraints. Services like real-time traffic signal information, hazard warnings, and pedestrian detection require low latency and high reliability due to their safety-critical nature. Conversely, services such as map updates and AI/ML model distributions can tolerate higher latency and variable bandwidth. To accommodate these varying requirements, V2I communications must support heterogeneous data flows. Implementing heterogeneous networks allows for the integration of multiple communication technologies, such as dedicated short-range communications (DSRC), cellular V2X (C-V2X), and mmWave communications. These technologies can be dynamically selected based on the service requirements. For instance, DSRC and C-V2X are suitable for low-latency communications needed for safety applications, while higher-frequency bands can support bandwidth-intensive services like HD map downloads and AI/ML model transmission. Hence, adaptive communication strategies are essential for managing network resources efficiently.

V2N communications connect vehicles to wider network infrastructures, primarily through cellular networks, enabling long-range data exchange and access to cloud services. Designed to support communication over extended distances, V2N surpasses the limitations of short-range V2I communications. This capability is vital for services that require connectivity beyond the immediate vicinity of the vehicle, such as cloud-based navigation, remote driving, and over-the-air updates. The adoption of Long-Term Evolution (LTE) and emerging 5G networks facilitates high data rates and broad coverage areas necessary for V2N communications. In particular, 5G networks offer enhanced capabilities with ultra-reliable low-latency communications (URLLC) and massive machine-type communications (mMTC), supporting the stringent requirements of autonomous driving services within the mobility DT.

I2N communications refer to the data exchange between roadside infrastructure components and the cloud server, typically over wired connections. Utilizing high-bandwidth, low-latency wired networks like fiber-optic cables or Ethernet connections, I2N communications are essential for transmitting large volumes of data collected by infrastructure components such as RSU cameras and LiDARs to central cloud platform within the mobility DT.

V2X communications are integral to the mobility DTs, enabling efficient data exchange among vehicles, infrastructure, and networks. By analyzing and implementing V2I, V2N, and I2N communications, the mobility DT can address the specific requirements and QoS of different mobility DT-enabled services.

2.2.3 Potential Standards

While there are currently no specific standards dedicated exclusively to digital twin technologies in smart mobility, existing standards in V2X communications and AI/ML model transfer provide a solid foundation for developing and evaluating mobility DTs. Specifically, the Third Generation Partnership Project (3GPP) has developed technical reports that address enhancements in 5G networks to support advanced V2X services and AI/ML capabilities. The two relevant standards are 3GPP technical report (TR) 22.886 entitled “Study on enhancement of 3GPP support for 5G V2X services” [22], and 3GPP TR 22.874 entitled “Study on traffic characteristics and performance requirements for AI/ML model transfer in 5GS” [142].

2.2.3.1 Relevance of 3GPP TR 22.886 and TR 22.874 to mobility DT

These standards are highly relevant to the mobility DT because they address the communication requirements and performance metrics essential for advanced V2X services and AI/ML functionalities, which are integral to the operation of a digital twin in smart mobility contexts. The mobility DT relies heavily on V2X communications to synchronize the physical and digital environments, enabling real-time data exchange and interaction among vehicles, infrastructure, and cloud services. The enhancements outlined in TR 22.886 provide essential insights into supporting advanced vehicular communications over 5G networks, ensuring that the mobility DT can operate effectively within the 5G ecosystem.

Similarly, TR 22.874 focuses on the requirements for transferring AI/ML models over 5G networks, which is pertinent to the mobility DT's need for updating and deploying AI/ML algorithms that underpin the DT's intelligent functions. Efficient AI/ML model transfer is crucial for maintaining up-to-date models in vehicles and infrastructure components, enabling adaptive learning and decision-making capabilities within the mobility DT.

By adhering to the specifications and recommendations outlined in these standards, mobility DT developers can ensure that their systems meet the necessary communication performance metrics, support advanced V2X services, and manage AI/ML model distribution effectively. This alignment facilitates interoperability, scalability, and reliability, which are critical for the successful deployment of the mobility DT in real-world smart mobility applications.

2.2.3.2 Introduction to 3GPP TR 22.886

3GPP TR 22.886 investigates potential enhancements required in 5G networks to support the next generation of V2X services, where four application groups are defined, i.e., vehicle platooning, advanced driving, extended sensors, and remote driving. Vehicle platooning allows vehicle to dynamically form a group, traveling in close proximity. Advanced driving supports semi-automated and fully-automated driving. Extended sensors defines the exchange of raw or processed sensor data between vehicles and other smart entities. Remote driving enables a remote driver or a V2X application to operate a remote vehicle. In each application group, various use cases are also defined for specific scenarios.

To support these advanced V2X services, TR 22.886 specifies performance requirements such as end-to-end (E2E) latency, reliability, data rates, and communication ranges. As shown in Table 2.2, we list four application groups with corresponding QoS requirements defined by 3GPP TR 22.886. The report also highlights the importance of edge computing integration, deploying computing resources at the network edge to reduce latency and support real-time data processing and decision-making.

By outlining these requirements and enhancements, TR 22.886 provides a framework that aligns with the communication and performance needs of the mobility DT. It ensures that the digital twin can effectively interact with the physical world through robust V2X communications, supporting the synchronization and real-time data exchange essential for advanced autonomous driving functionalities.

2.2.3.3 Introduction to 3GPP TR 22.874

3GPP TR 22.874 examines the network requirements for transferring AI/ML models over 5G networks. As AI and ML become increasingly integral to applications like autonomous driving and digital twins, efficiently distributing and updating models is critical.

The report identifies several use cases relevant to AI/ML model transfer. One such use case is AI/ML model/data distribution and sharing, where the system may continuously exchange and share AI/ML model layers in a distributed or federated network determined by a change of events, conditions, or emergence situations. Another significant use case is federated learning, a collaborative learning approach where edge devices train models locally using their data and periodically upload model updates to a central server. The server aggregates these updates to refine the global model, which is then redistributed to the edge devices. This method reduces the need to transfer raw data, preserving privacy and reducing network load, which is particularly important for handling the vast amounts of data generated by autonomous

Table 2.2: QoS requirements for application groups defined in 3GPP TR 22.886 [22]

Application groups	Latency (ms)	Reliability (%)	Data rate (Mbps)	Communication range (m)
Vehicle platooning	10∼25	90∼99.99	50∼65	80∼350
Advanced driving	3∼100	99.99∼99.999	10∼53	360∼700
Extended sensors	3∼100	90∼99.999	10∼1000	50∼1000
Remote driving	20	99.999	Uplink: 25 Downlink: 1	Ubiquitous

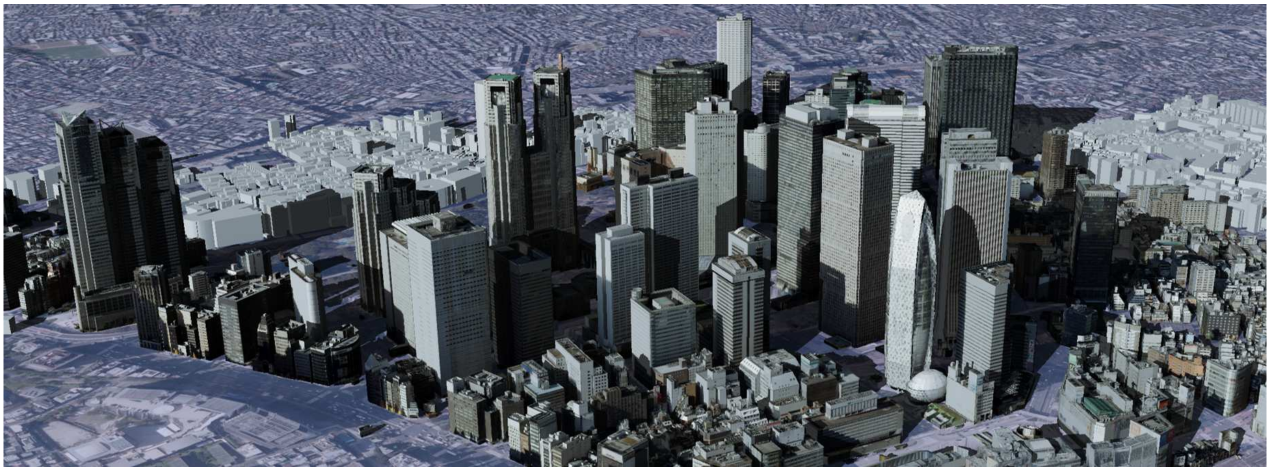
vehicles.

TR 22.874 analyzes the traffic characteristics associated with AI/ML model transfers, noting that model sizes can range from a few megabytes to several gigabytes, depending on complexity. The frequency of updates can vary from real-time to periodic, based on application needs. The report also specifies network performance requirements, including sufficient data rates to transfer large models within acceptable time frames, E2E latency for real-time updates, and high reliability to ensure successful model transfers.

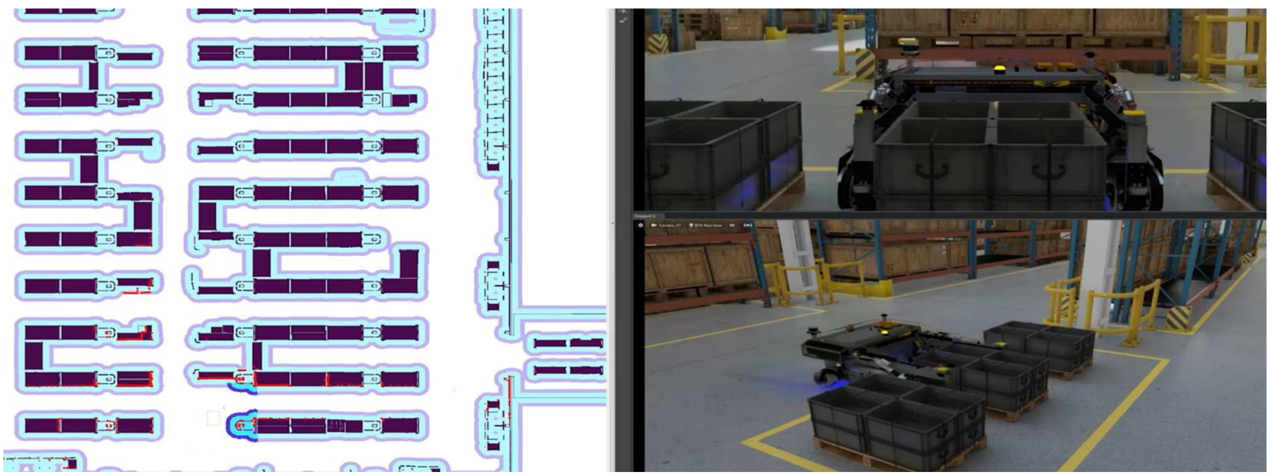
For the mobility DT, TR 22.874 is particularly relevant because it addresses the challenges associated with distributing AI/ML models to vehicles and infrastructure components. Efficient model transfer is essential for keeping the DT updated with the latest algorithms, ensuring consistent performance across the system, and enabling adaptive learning and decision-making capabilities. By adhering to the recommendations in TR 22.874, developers can optimize the mobility DT's AI/ML functionalities, leveraging the advanced capabilities of 5G networks to enhance the system's overall intelligence and responsiveness.

2.3 Challenges Towards Smart Mobility Digital Twin

The development of the SMDT presents numerous challenges that span technological, organizational, and societal domains. The term "smart" in SMDT refers to the seamless integration of highly/fully autonomous vehicles (L4/L5) into a mobility DT, enabling real-time decision-making, enhanced coordination, and optimized traffic flow. While the mobility DT holds the potential to revolutionize transportation systems by providing a dynamic virtual replica of physical mobility networks, realizing the term "smart" requires overcoming significant hurdles Moreover, SMDT involves not only bridging DTs and autonomous systems but also the modeling of traffic and urban systems, both of which are critical to achieving a comprehensive and robust implementation. While some existing technologies like the Omniverse DT [31] demonstrate their transformative role in advancing autonomous systems and platforms like the Tokyo DT project [32] showcase the potential of DTs to optimize urban planning and mobility management, combining these approaches into a unified SMDT framework presents significant challenges. Figure 2.9(a) and Figure 2.9(b) show Omniverse DT-based robotics simulation toolkit and 3D models of Tokyo city in the Tokyo DT project, respectively. This section analyzes the primary challenges impeding the implementation of SMDT, including comprehensive system design, use case definition, lack of proof of concept (PoC), data inter-



(a) 3D models of Tokyo city from the Tokyo DT project



(b) Omniverse DT-based robotics simulation toolkit

Figure 2.9: Existing smart mobility-related DT technologies [31,32].

operability, security and privacy concerns, and scalability issues.

1. *Comprehensive system design:* Designing a comprehensive mobility DT system is inherently complex due to the need to integrate diverse components such as vehicles, infrastructure, communication networks, and data analytics platforms. Achieving seamless interoperability among these components requires standardized interfaces and protocols, as well as robust architectures that can accommodate heterogeneous technologies. The challenge lies in creating a unified framework that can support real-time data exchange, processing, and decision-making across various subsystems while ensuring reliability and resilience against failures.

2. *Use case definition:* Clearly defining use cases for the mobility DT is crucial for guiding development efforts and demonstrating value to stakeholders. However, identifying and prioritizing use cases is challenging due to the multifaceted nature of smart mobility, which encompasses aspects like traffic management, autonomous driving, public transportation optimization, and environmental monitoring. Each use case may have different requirements and constraints, making it difficult to develop a one-size-fits-all solution. Moreover, aligning the interests of diverse stakeholders, including government agencies, private companies, and end-users, adds to the complexity of use case definition.

3. *Lack of PoC:* The absence of comprehensive PoC implementations hampers the validation of mobility DT technologies and their potential benefits. Without practical demonstrations, it is challenging to assess the feasibility, performance, and return on investment of mobility DT initiatives. This lack of tangible evidence makes it difficult to secure funding, gain stakeholder buy-in, and drive widespread adoption. Developing pilot projects and testbeds is essential but requires significant resources, coordination, and risk-taking from involved parties.

4. *Data interoperability and standardization:* Data interoperability is a significant challenge due to the variety of data formats, protocols, and standards used by different systems and devices within the smart mobility ecosystem. Ensuring that data from various sources—such as sensors, vehicles, and infrastructure—can be seamlessly integrated and interpreted requires the adoption of common data models and standardization efforts. The lack of widely accepted standards leads to fragmentation and hinders the ability to aggregate and analyze data effectively on a large scale.

5. *Security and privacy concerns:* The mobility DT relies heavily on the continuous exchange of vast amounts of data, some of which may be sensitive or personal. Protecting this data from unauthorized access, breaches, and cyber-attacks is a paramount concern. Implementing robust security measures is challenging due to the distributed nature of the system and the resource constraints of edge devices.

6. *Scalability and performance issues:* Scalability is a critical challenge as the mobility DT must handle increasing amounts of data and a growing number of connected devices without degradation in performance. Processing real-time data streams from millions of sources requires substantial computational and networking resources. Designing sys-

tems that can scale horizontally and vertically while maintaining low latency and high throughput is complex. Furthermore, optimizing resource allocation between cloud and edge computing environments to support real-time analytics and decision-making adds another layer of difficulty.

Addressing these challenges is essential for the successful implementation of the SMDT. Overcoming them will require collaborative efforts among researchers, industry practitioners, policymakers, and other stakeholders. In later technical Chapters. we will focus more on system design, defining use cases, and establishing PoC experiments.

2.4 Summary

In this Chapter, we first survey the SOTA autonomous driving technologies, including navigation, environmental perception, localization, prediction, planning, and motion control. Then some concepts and techniques in the establishment of mobility DTs, such as edge/cloud computing, V2X communications, and standards, are introduced. Finally, we point out the challenges ahead of the successful transition from mobility DT to SMDT.

Chapter 3

Smart Mobility Digital Twin: System Design and Implementation

This Chapter introduces the smart mobility digital twin (SMDT) platform, according to the comprehensive introduction of mobility DT establishment methods in Chapter 2. The platform bridges the gap between physical and cyber worlds by utilizing cloud and edge computing resources, enabling real-time traffic monitoring, decision-making, and system optimization. First, in Section 3.1, we provide the motivation for introducing SMDT to the autonomous driving field. In Section 3.2, we present the high-level conceptual architecture of the SMDT platform, detailing its integration of RSU edge, CAV edge, and a central cloud. In Section 3.3, we describe the prototype system and its deployment in a real-world environment, highlighting the hardware and software components used to create real-time traffic DTs. Finally, we summarize this Chapter in Section 3.4.

3.1 Motivation

Digital twins (DTs) enable a synergistic integration between the physical and cyber worlds, thereby becoming a catalyst for the ongoing digital transformation of our society [143]. In essence, the continuous evolution of DTs over the past two decades signifies their underlying potential to lead this digital transformation [144]. In particular, DTs possess remarkable abilities to establish dynamic digital models of the environment and bidirectional communications between the physical and cyber worlds. Thereby, many functionalities such as real-time monitoring, design, and optimization can be developed with DTs acting as their founding basis.

The remarkable advancements have driven revolutionary applications into various sectors like manufacturing, agriculture, and transportation [145,146].

At the forefront of these viable sectors, integrating DTs into the transportation and traffic fields constitutes a promising solution for critical challenges like traffic congestion and road accidents [33]. As discussed in Chapter 1, global DT market reached USD 16.75 billion in 2023 and was predicted to grow at a compound annual growth rate (CAGR) of 35.7% until 2030 [34]. Notably, the automotive and transport segment has dominated the DT market in 2023, accounting for over 21% of its total revenue. This remarkable market performance highlights the crucial role of DTs as a cornerstone in the smart mobility infrastructure [147,148].

Indeed, the convergence of smart mobility systems with DTs has ushered in new promising levels of autonomous driving breaking into connected and automated vehicles (CAVs), thereby introducing the concept of parallel intelligence [149,150]. On the one hand, having these vehicles "connected" underpins the feasibility of DT deployment. In particular, vehicles outfitted with vehicle-to-everything (V2X) can share information with various external entities by using vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-network (V2N) communications [56]. In particular, V2N facilitates vehicle data transmission to the cloud, naturally enabling the establishment of DTs on the cloud plane [35]. On the other hand, the performances of onboard autonomous driving systems are hindered by multiple constraints, such as onboard sensing view and computing resources [151]. Henceforth, incorporating DT technologies or parallel intelligence is anticipated to overcome such challenges that have hindered the evolution of CAVs.

Despite these promising strides, a significant gap persists in validating these advantages upon integrating DTs in an end-to-end (E2E) real-world smart mobility. Indeed, the theoretical advantages of DTs, as demonstrated in [152,153,154], still require practical validation to ascertain their feasibility and impact. the overarching goal of this Chapter is to propose a novel SMDT platform that can be used to establish a real-world and real-time traffic DT, to enhance CAVs' ability to "see more and see further" in autonomous driving scenarios.

To validate the feasibility of applying the SMDT platform within the domain of autonomous driving and explore its potential, the most important task is the system design and implementation in the real world, which poses numerous challenges. Firstly, it is essential to create a real-time virtual representation of real-world traffic environments (i.e., traffic DT), which involves sensing, perceiving, and visualizing various static and dynamic entities. Secondly, the system design should consider safety and robustness issues, because of mas-

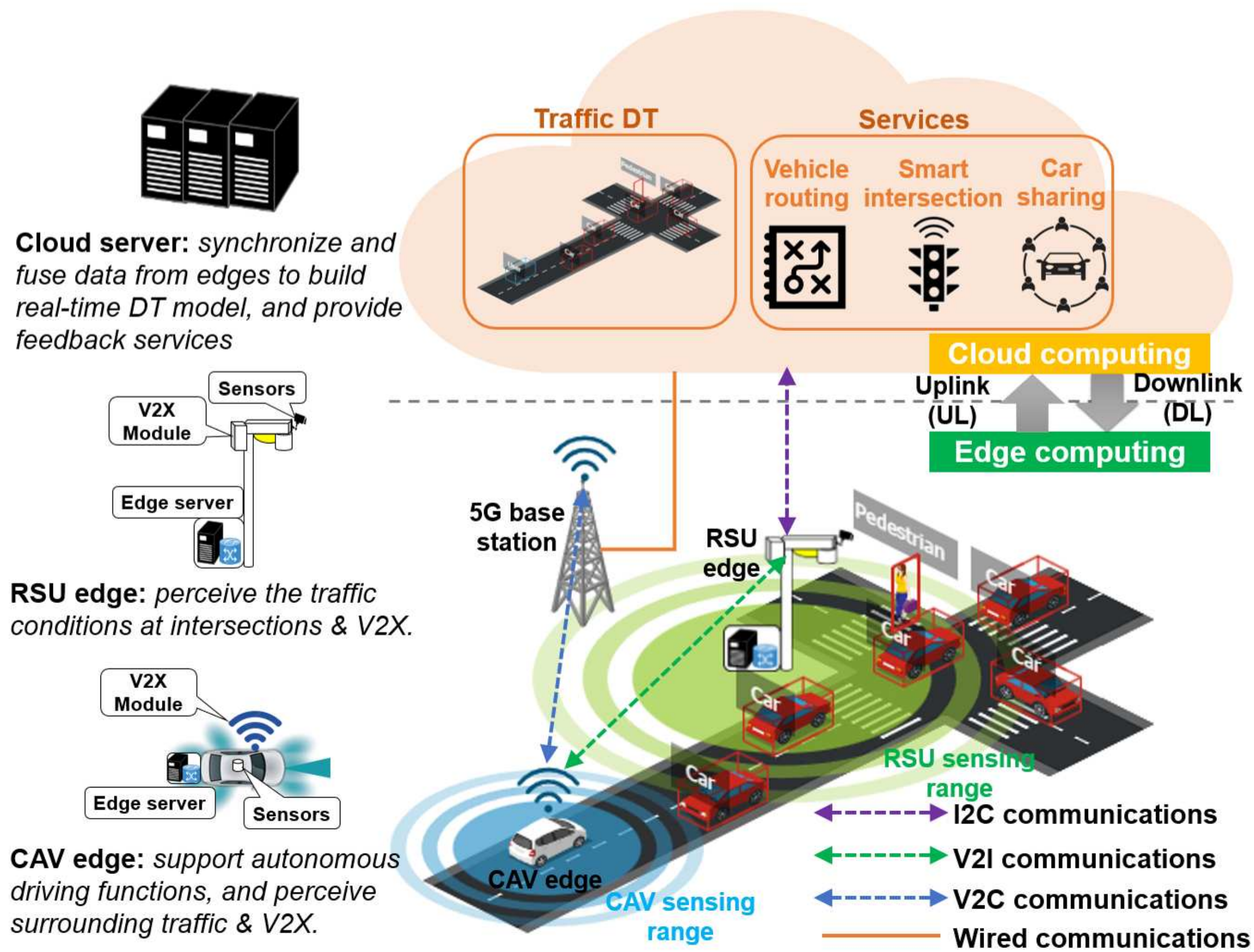


Figure 3.1: SMDT high-level conceptual system architecture with cloud/edge computing.

sive data collection from traffic environments and the requirement for autonomous vehicles to operate in unpredictable conditions, which necessitates an effective solution for dynamic traffic information acquisition and a reliable V2X communication network. Finally, successful implementation requires a holistic consideration and integration of hardware, software, and communication, as well as collaboration between cloud and edge computing.

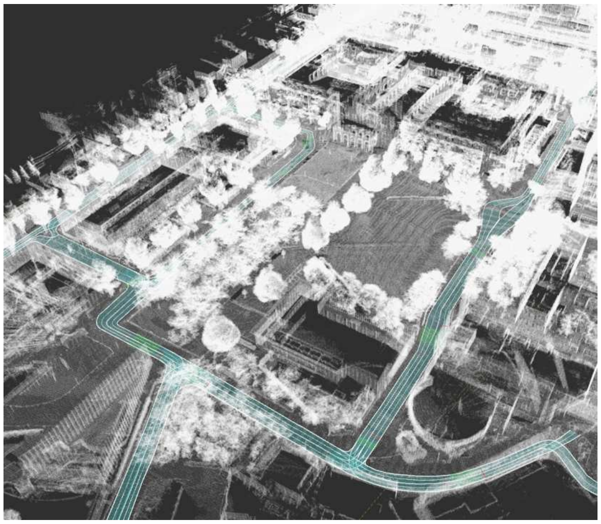
3.2 SMDT Platform Architecture

The proposed SMDT platform integrates the cloud and edge computing resources of intelligent transportation systems (ITS). This architecture is chosen to exploit the robust computational capabilities of the cloud for data-intensive tasks while utilizing edge computing to minimize response times for real-time services. This dual-computing approach is essential to manage latency, safety, and processing efficiency in large-scale traffic network operations effectively. Computing at the cloud can utilize plentiful and stronger processing units, but it introduces considerable network latency. The SMDT platform prioritizes cloud computing when handling computation-intensive tasks that require global big data from large-scale traffic networks. Computing at the edge cannot enjoy the same processing efficiency as cloud computing due to the limited power supply and space for deploying units. However, it guarantees real-time responses to the users, addressing their demands for delay-sensitive services on the SMDT platform. As depicted in Figure 3.1, the computing units of the SMDT platform are categorized into three types: RSU edges, CAV edges, and a central cloud. They are interconnected and collaboratively support the platform operations, including *traffic DT* creation, data processing, and user interaction.

1) RSU edge: RSUs play an important role in the SMDT platform, primarily focused on high-traffic areas. RSUs are equipped with a variety of sensors to monitor traffic flow and detect accidents, and they also have the capability to receive data from connected vehicles (CVs) through V2I communication. To ensure their effectiveness, RSUs are strategically deployed at traffic hotspots characterized by high traffic density, complex flows, and higher accident rates. This targeted deployment ensures that RSUs are placed where they can have the greatest impact on managing traffic efficiently, enhancing safety, and optimizing flow. These two roles of RSU not only enhance the effectiveness of the *traffic DT* but also optimize their utilization in the overall network.

2) CAV edge: All CAVs operate in a highly automated mode and strictly adhere to directives issued by the cloud. Besides contextual awareness, CAV sensors are used for localization, local path planning, and motion control. As a basic requirement of the SMDT platform, CAVs should report their positions to get services and upload their sensing data to compensate for the undetectable areas of RSU sensors when constructing the *traffic DT*. In essence, CAVs are the mobile counterparts of RSUs. Importantly, these CAVs should be designed with a monitoring system that can be aware of the signal loss and switch to standalone mode for decision-making, ensuring that even if a CAV temporarily loses its communication link, it can still maintain safety and operational integrity until the connection is re-established.

3) Central cloud: The cloud is the place where a dynamic *traffic DT* is established by integrating a comprehensive array of traffic data. This integration encompasses real-time information from both RSU and CAV edges, along with static build-in data like high-definition



(a) An example of HD map.

(b) An example of static 3D models.

Figure 3.2: Built-in data in the central cloud.

(HD) maps and inputs from various commercial traffic information providers. As shown in Figure 3.2, we show an example of HD maps (including point cloud map and road vector map) and 3D models of our testing field. Serving as a vital resource pool, the cloud processes and analyzes this diverse data to offer a detailed and expansive view of traffic conditions. Such a robust integration of multiple data sources is instrumental for both providing accurate traffic analysis and ensuring the robustness of the system, thereby enhancing management solutions for road safety and traffic efficiency.

Data sharing between cloud and edge computing can be divided into downlink (DL) and uplink (UL). The edge servers at the RSU and CAV continuously upload various levels of real-time sensing data, such as raw data or processed data, according to different use cases. The cloud provides services and issues directives over the DL to edges. To ensure connectivity among these smart entities and reliable DL/UL between cloud and edge, it is necessary to establish a V2X network, including V2N, V2I, and infrastructure-to-network (I2N) communications, where V2N communication is designed with a multi-radio access technology (Multi-RAT) handover mechanism based on channel conditions to ensures consistent and stable communication. V2N communication ensures the reach to the cloud through cellular networks or relayed by RSUs using dedicated short-range communications (DSRC), millimeter-wave (mmWave), and Wi-Fi.

3.3 Prototype System and Outdoor Implementation

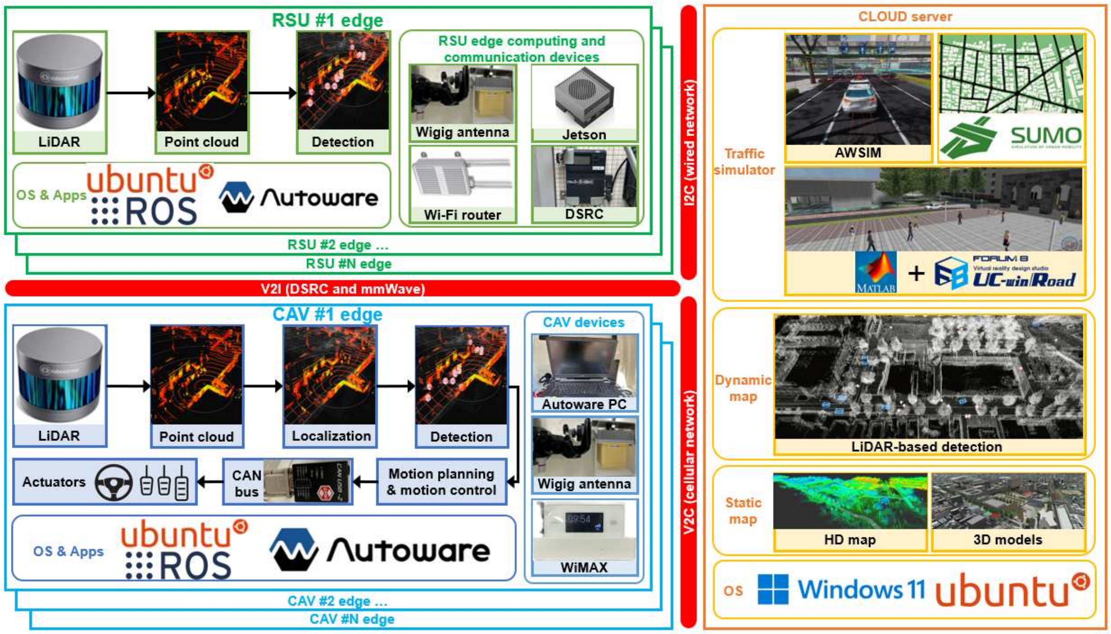
In this section, we design an example system architecture of the proposed SMDT platform tailored for implementation, as shown in Figure 3.2(a). In the vein of conceptual design, the example design also includes three key components: RSU, CAV, and the central cloud. The primary objective of this implementation-tailored system is to realize the conception of SMDT platform depicted in Figure 3.1, utilizing LiDAR sensing, cloud-edge computing, and V2X communication.

3.3.1 Hardware Deployment and Heterogeneous V2X Network

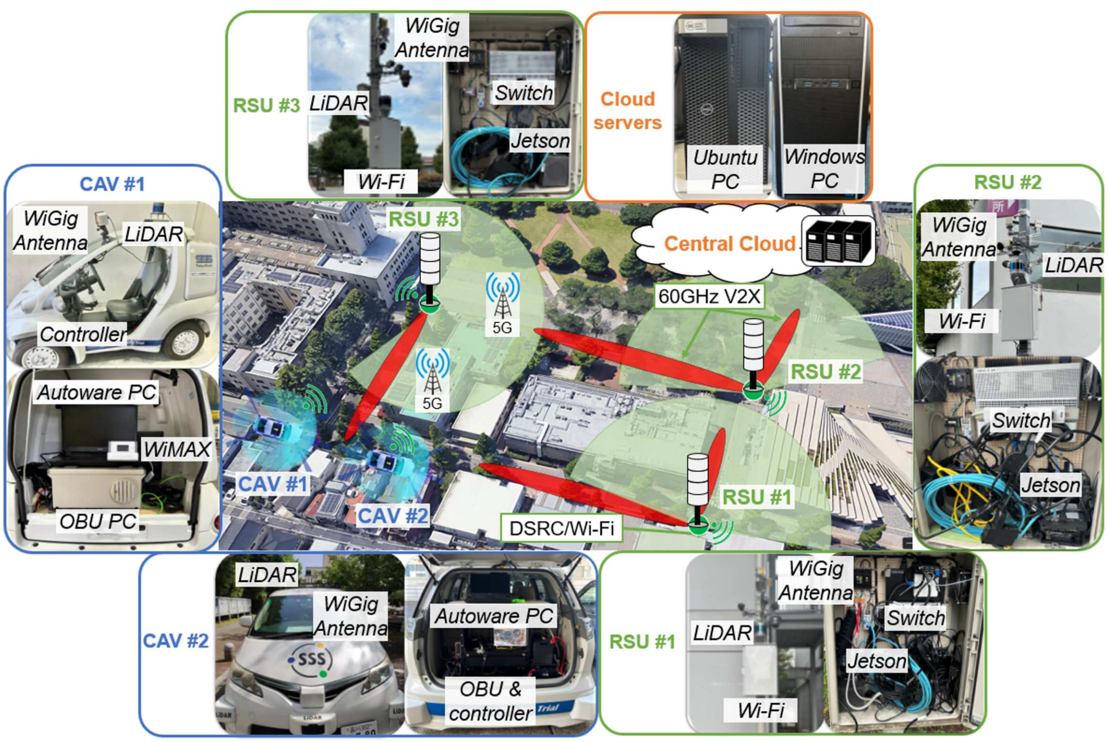
Our SMDT testing environment is named Sci Tokyo Smart Mobility R&E Field[1], as shown in Figure 3.2(b). In the field setting, there are three RSUs installed at three corners of a square road section, which represent typical examples of complex urban traffic scenarios. This deployment strategy enables comprehensive monitoring of vehicle flows within the square region enclosed by these intersections. Additionally, the testing field includes two CAVs capable of high autonomy and a remote cloud server located far away from the field. The communication system within the SMDT platform is based on a heterogeneous V2X network and managed by a software-defined network (SDN) controller [52], which enables three kinds of V2X links: V2I, V2N, and I2N communications. Considering the large communication distance between vehicles and the cloud, the cellular network is applied for V2N communication. Depending on the application scenario, RSUs may send data of different levels to the cloud server and nearby vehicles, such as raw data or processed data, which have distinct requirements for communication coverage and bandwidth. Hence, we use wired networks for I2N communication and employ both DSRC and mmWave technologies for V2I. Hence, each RSU and CAV are equipped with communication modules, including WiMAX for V2N communication, a Wi-Fi router as a replacement for DSRC, WiGig antenna for mmWave communication, and Ethernet cables for wired communication through the campus network. In Table 3.1, we list the required communication speeds and coverage in our experiment field, as well as the performance of selected communication methods.

As for sensing and computing devices, summarized in Table 3.2, an 80-layer LiDAR is integrated alongside an NVIDIA Jetson in each RSU. LiDAR is responsible for capturing raw data (i.e., dynamic point clouds) from the physical world. The Jetson, on the other hand,

[1] https://www.wise-sss.titech.ac.jp/en/.



(a) Example system design of the SMDT platform.



(b) Hardware components in the testing field.

Figure 3.3: SMDT platform: Example system design and implementation.

is employed to implement specific functional modules, such as object detection and tracking based on point clouds, thereby enabling object-level projection from the physical world to the digital world. Two CAVs are outfitted with 32-layer LiDAR sensors positioned on the rooftop to sense their surroundings. A dedicated Autoware PC is utilized for processing the LiDAR data, facilitating environmental perception, localization, planning, and motion control. The control signals are then transmitted to the onboard unit (OBU) to achieve autonomous driving. The remote cloud server works as the central hub for data aggregation and storage, global information processing, and providing feedback services to the autonomous vehicle. Therefore, the performance requirements for the cloud server are very high, including scalability, processing power, and reliability to efficiently and securely handle large volumes of real-time data. Consequently, we select a computer equipped with a GeForce RTX 5000 GPU and ensure abundant memory and storage resources to satisfy the computational demands.

Table 3.1: Communication Requirements and Performances

<table>
<tr><th rowspan="2">V2X modality</th><th colspan="2">Requirements</th><th rowspan="2">Information contents</th></tr>
<tr><th>Methods</th><th>Performances</th></tr>
<tr><td rowspan="2">V2N</td><td colspan="2">Data rate: ≥ 80 Mbps
Coverage: ≥ 1 km</td><td rowspan="2">Uplink: vehicle position and processed sensor data.

Downlink: cloud-based decisions</td></tr>
<tr><td>WiMAX</td><td>≥ 120 Mbps
≥ 50 km</td></tr>
<tr><td rowspan="4">V2I</td><td colspan="2">Data rate: ≥ 1 Mbps
Coverage: ≥ 50 m</td><td rowspan="2">Cooperative perception (processed data, e.g., objection detection, object tracking, etc.</td></tr>
<tr><td>Wi-Fi router</td><td>≥ 10 Mbps
≥ 200 m</td></tr>
<tr><td colspan="2">Data rate: ≥ 1 Gbps
Coverage: ≥ 20 m</td><td rowspan="2">Cooperative perception (raw sensor data, e.g., LiDAR point cloud and camera images)</td></tr>
<tr><td>MmWave
Wigig</td><td>≥ 1 Gbps
≥ 120 m</td></tr>
<tr><td rowspan="2">I2N</td><td colspan="2">Data rate: ≥ 1 Gbps</td><td rowspan="2">Environmental perception (both raw data and processed data)</td></tr>
<tr><td>Ethernet</td><td>≥ 1 Gbps</td></tr>
</table>

3.3.2 Software Installation and Traffic DT Modeling

The software in our platform is centered around Autoware [155] and robot operating system (ROS). Autoware facilitates the detection and tracking of traffic participants within the RSU sensor range. The detection module applies the CenterPoint framework [93], which can detect, identify, and visualize 3D objects from the LiDAR point clouds in real time. Then, a multiple object tracking (MOT) module [97], is responsible for assigning the detected objects with IDs and estimating their velocities. An example of detection and tracking results is shown in Figure 3.4. Additionally, a normal distributions transform (NDT) algorithm is utilized for the LiDAR scan matching with the 3D point cloud map, which enables real-time localization with centimeter-level accuracy. The driving route is represented in the form of waypoints,

Table 3.2: Hardware Devices and Specifications

Hardware	**Device name**		**Specifications**
	Location	**Amount**	
CAV	RoboCar (CAV #1)		L4/L5 autonomous vehicle with modular ADS, onboard sensors, and vehicle controller.
	-	1	
	MiniVan (CAV #2)		
	-	1	
Sensor	RS-LiDAR-32		Range: 200 m, Accuracy: ± 3 cm, Rotation Speed:10/20 Hz
	CAV edge	2	
	RS-LiDAR-80		Range: 200 m, Accuracy: ± 3 cm, Rotation speed: 5/10/20 Hz
	RSU edge	3	
Cloud/ edge servers	Autoware PC #1		OS: Ubuntu 16.04, ROS: Kinetic, Autoware. AI
	CAV #1 edge	1	
	Autoware PC #2		OS: Ubuntu 22.04, ROS: Foxy, Autoware. Universe
	CAV #1 edge	1	
	Jetson AGX Orin		OS: Ubuntu 20.04 (JetPack 5.0.2), ROS: Galactic, Autoware. Universe
	RSU edge	3	
	DT engine		OS: Ubuntu 20.04, ROS: Galactic, Autoware. Universe
	Cloud	1	
	DT simulator		OS: Windows 11, Simulation app: SUMO, UC-win/Road
	Cloud	1	

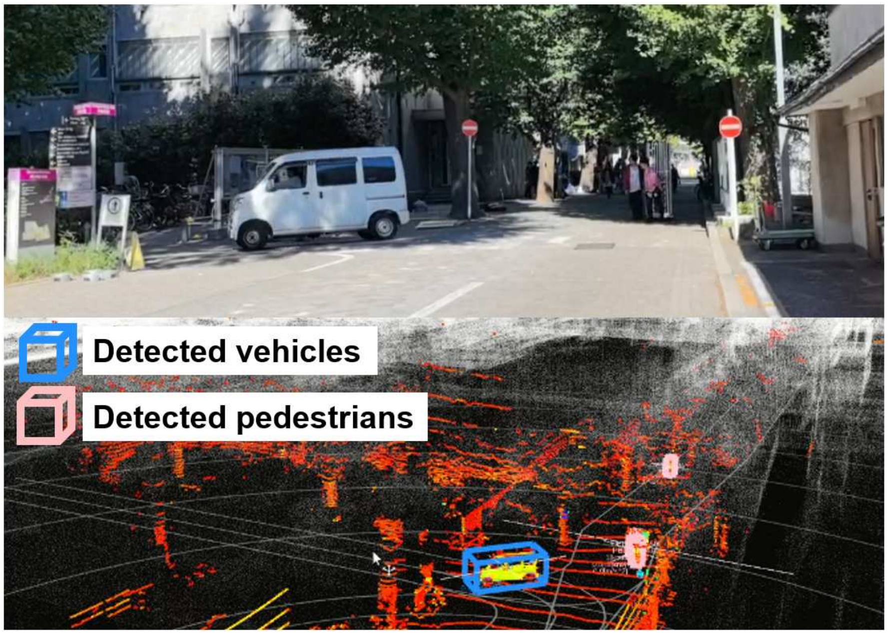


Figure 3.4: An example of LiDAR-based object detection.

which includes a set of points with coordinates and desired speed. To perform local motion planning and motion control, a velocity planner adjusts the velocity based on the waypoints to decelerate or accelerate in response to nearby objects and road characteristics, such as stop lines and traffic lights. Finally, a pure pursuit algorithm is employed to generate coordinated velocities and steering angles and follow the target waypoints.

Over the cloud plane, since ROS facilitates seamless communication among distributed computers within a local area network (LAN), the cloud server can easily access all ROS2-defined messages from RSU edges. Besides, the hypertext transfer protocol (HTTP) communication is utilized to establish the V2N link, enabling the real-time detection results (a kind of ROS2-defined massages) uploading and planned route downloading on CAV edges. After acquiring detection results from RSUs and CAVs, we can fuse them and build the real-time object-level *traffic DT* in Autoware visualization tool Rviz, as shown in Figure 3.5. To fuse the LiDAR detection results from RSUs and CAVs, a late fusion method is applied, which assumes that the detection accuracy of each RSU is extremely high. First, the overlapping regions of the LiDAR sensing ranges are estimated based on the fixed positions of RSUs and CAVs. Within these overlapping regions, objects detected by different LiDARs are compared. For each object in the detection results of one LiDAR, its distance to objects detected by other LiDARs is computed. If the distance between two objects is less than 0.5 meters and the ob-

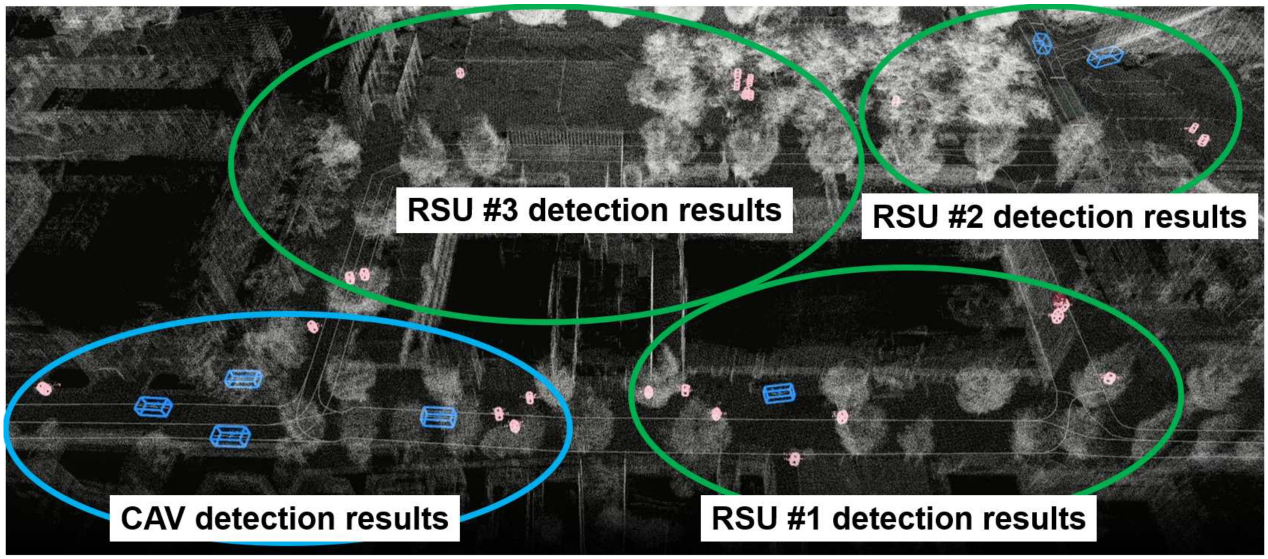


Figure 3.5: An illustration of traffic DT modeling in testing field.

ject types match, the two objects are merged by averaging their positions and attributes. Unprocessed objects outside the overlapping regions or unmatched objects are retained in their original form. This fusion approach ensures accurate and robust integration of data from multiple LiDARs, enabling a comprehensive and consistent view of the environment. Then, the cloud server utilizes objects' positions to align them with the corresponding road segments, which facilitates traffic monitoring on each road section, thereby realizing cloud-based decisions or services for CAVs. Additionally, for large-scale traffic experiments that cannot be executed within our testing field, we also create the simulation environment with some traffic simulation software, such as SUMO [156] and UC-win/Road [157].

3.4 Summary

This Chapter outlines the design and implementation of the SMDT platform, which bridges the physical and cyber worlds to enhance autonomous driving capabilities. The Chapter begins by motivating the need for SMDT in addressing challenges like limited onboard resources, traffic congestion, and safety concerns. It emphasizes the potential of SMDT to integrate cloud and edge computing for real-time traffic monitoring, decision-making, and system optimization. The architecture of the SMDT platform is introduced, detailing its three core components: RSU edge, CAV edge, and a central cloud. These components collaborate to create and manage real-time traffic DTs, leveraging V2X communication to ensure efficient data exchange and system responsiveness. The integration of cloud computing enables handling large-scale traffic data, while edge computing ensures low-latency responses for time-sensitive tasks. Then, we further describe the prototype system and its implementation in the Sci Tokyo Smart Mobility R&E Field. Key hardware elements include LiDAR sensors, NVIDIA Jetson, and high-performance cloud servers. The software framework is based on Autoware and ROS, supporting functionalities like real-time object detection, tracking, and route planning. Finally, we show the establishment of traffic DT in our testing field, demonstrateing the platform's potential to improve traffic efficiency, enhance road safety, and enable scalable deployment in real-world scenarios, marking a significant step toward the realization of smart mobility systems.

Chapter 4

Global Digital Twin-Enabled Autonomous Vehicle Navigation System

This Chapter presents the design and implementation of a connected and automated Vehicle (CAV) navigation system based on the smart mobility digital twin (SMDT) platform proposed in Chapter 3. The CAV navigation system aims to enhance traffic management by integrating real-time dynamic traffic data, vehicle-to-everything (V2X) communication, and advanced route planning mechanisms. First, in Section 4.1, we outline the motivation for developing a navigation system tailored to address the limitations of existing autonomous vehicle frameworks. In Section 4.2, we introduce the high-level design of the CAV navigation system, emphasizing its event-triggered route planning strategy and real-time adaptability. In Section 4.4, we demonstrate a proof-of-concept (PoC) implementation within the Sci Tokyo Smart Mobility Field, showcasing the system's ability to respond to real-world traffic events. In Section 4.5, we evaluate the system's performance through large-scale simulations, focusing on traffic efficiency, safety improvements, and communication metrics. Finally, we summarize this Chapter in Section 4.6.

4.1 Motivation

As urban populations grow and vehicle ownership rises, urban driving has become increasingly challenging and stressful due to congested traffic conditions, making it one of the most suffering aspects of modern city life. Hence, efficient traffic management has emerged as a critical objective for metropolitan areas worldwide. To alleviate these issues, numerous commercial GPS navigation applications, such as Google Maps [83] and Baidu Maps [84], have been developed to assist drivers in navigating through complex urban landscapes by providing optimal routes and real-time traffic updates.

In the realm of autonomous driving, navigation systems have also witnessed significant advancements in recent years. Contemporary frameworks like OpenPlanner focus on calculating the shortest possible paths based on static high-definition (HD) maps [88]. However, unlike navigation systems designed for human drivers, which dynamically integrate real-time traffic conditions and historical data to optimize routes, autonomous vehicle navigation frameworks often lack the capability to incorporate such dynamic traffic information into their route planning [158,159]. This limitation stems from the complexity involved in integrating real-time sensor data, V2X communications, and centralized traffic updates into the onboard decision-making processes of autonomous vehicles [160].

The proposed SMDT platform in Chapter 3 addresses this gap by providing access to real-time global traffic information. Leveraging the SMDT, we aim to optimize traffic management and fully exploit its potential to enhance the travel efficiency of autonomous vehicles. This motivation drives the development of a navigation system that not only calculates optimal routes but also adapts dynamically to changing traffic conditions by integrating real-time data from various sources.

The overarching goal of this Chapter is to propose a global DT-enabled navigation system for CAVs. This system leverages the capabilities of the SMDT to provide autonomous vehicles with real-time, holistic traffic insights, enabling more efficient and adaptive route planning. By bridging the gap between static map-based navigation and dynamic traffic-aware routing, we seek to improve the overall efficiency and reliability of autonomous vehicle operations in urban environments.

4.2 Design of CAV Navigation System

To evaluate and highlight the benefits of the proposed SMDT platform from the practical viewpoint of applications and services, a CAV navigation system is designed based on SMDT. The aim of the CAV navigation system is to provide the most efficient global path for users navigating the traffic network, as shown in Figure 4.1. In a dynamic traffic environment,

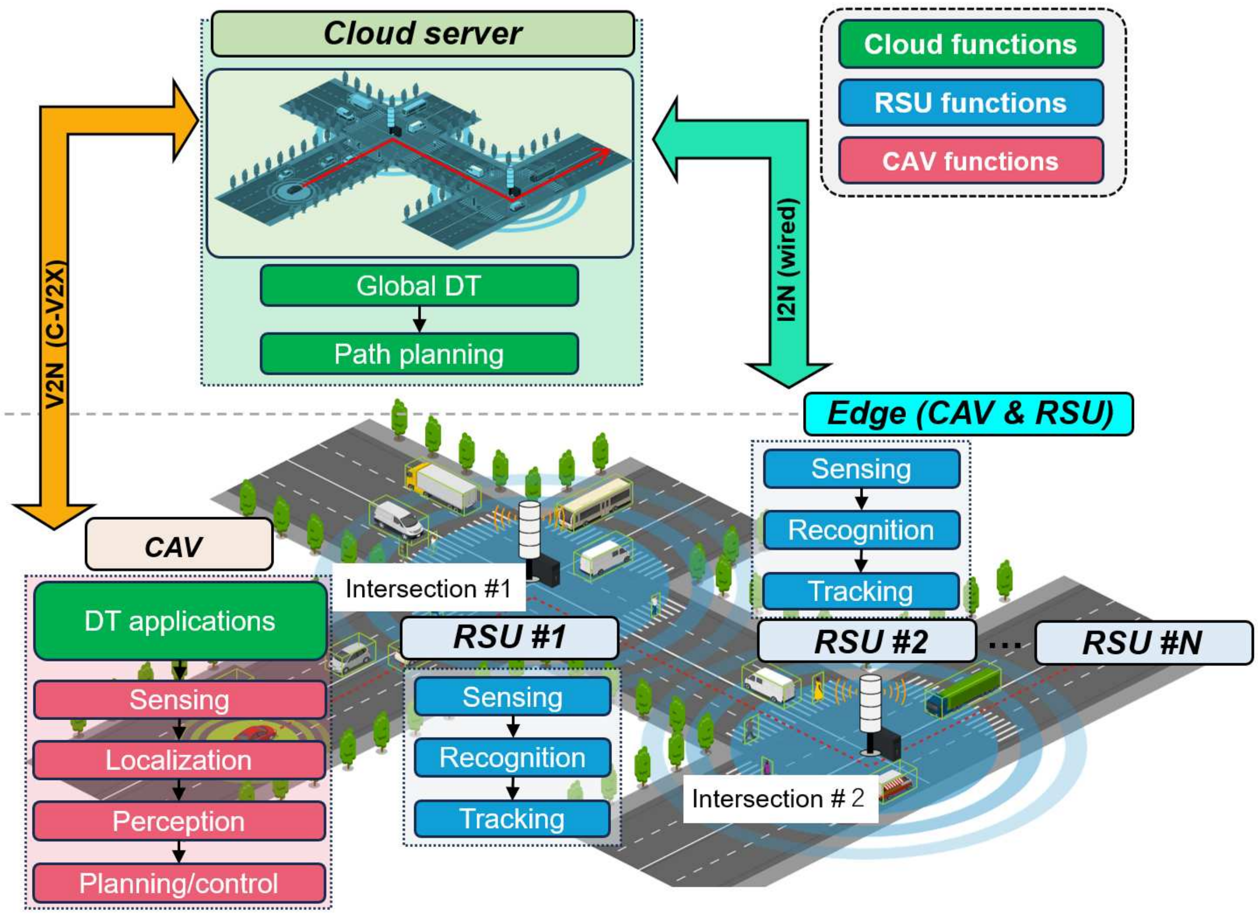


Figure 4.1: High-level Conceptual Design of CAV Navigation System.

road segments may experience unexpected incidents, such as accidents, large gatherings, or peak-time congestion that can drastically affect journey time and road safety. Therefore, it is crucial to devise a mechanism that continuously detects such incidents and dynamically re-routes users.

Hence, the CAV navigation system is designed to employ an event-triggered planning strategy. As users enter the road network, the optimal routes are generated based on current traffic conditions to minimize journey time from origin to destination. During transit, if specific events occur within the traffic network, such as a car crash, a route re-planning service is triggered to help users avoid these events. Overall, the CAV navigation system should offer the following capabilities:

- *Journey time calculation:* Upon a user's entry into the traffic network, it is essential to furnish them with an initial planned route connecting their origin and destination, and thus it is necessary to estimate the journey time of each road segment.

- *Event-triggered mechanism*: If there exists an overlap between the planned route and road segments with unexpected events, a re-planned path must be generated to circumvent the impacted areas.

4.2.1 Journey Time Calculation

To accurately estimate the journey time for each road segment, a multitude of factors, both time-variant and time-invariant, need to be taken into consideration. Time-invariant factors include road distance, road width, and free-flow speed, while time-variant factors encompass signal timing, traffic volume, and the temporal evolution of traffic flow. To facilitate a tractable model for journey time computation, this Chapter operates under two foundational assumptions: *non-signalization* and *dynamic equilibrium*. Since cooperative driving in non-signalized intersections can increase traffic safety and improve traffic flow and thus has become very popular in recent years [161], we postulate that all nodes in the traffic network are non-signalized. As such, the waiting times induced by traffic signals can be disregarded. Additionally, we posit that the traffic network exists in a state of dynamic equilibrium, which is a common supposition in related traffic equilibrium research [162,163]. Under this condition, the inflow and outflow rates for each road segment achieve a balanced state, yielding relatively stable traveling speeds and journey times. Consequently, the route planning strategy can be characterized as deterministic, which means when we need to make a route-planning decision at time t, the traffic condition at time t can be used to calculate the journey time of each road segment and find the fastest path for the user. Under the *non-signalization* and *dynamic equilibrium* assumptions, the traffic volume is the primary time-variant factor that needs to be considered in constructing the journey time estimation model.

We use a directed graph $G = (N, L)$ to represent the traffic network. N is the set of nodes (i.e., traffic intersection) and $N = \{n^i\}_{i=1}^{M}$, where M is the total number of nodes. L is defined as the set of directed links (i.e., unidirectional traffic roads) and $L = \{l^{i,j} = (n^i \rightarrow n^j), n^i, n^j \in N\}$, where l^{ij} means the directed road from n^i to n^j. Assuming that we can obtain the real-time traffic volume x on the road link l, based on the traffic volume and road length s, we can calculate the traffic density k with:

$$k = \frac{x}{s} \tag{4.1}$$

According to [164], the relationship between traffic density k and journey speed v can be modeled as:

$$v = v_{\text{free}}(1 - \frac{k}{k_{\text{max}}}) \tag{4.2}$$

where v_f is the free-flow speed, and k_{max} is the maximum vehicle density. Thus, the journey time can be calculated as:

$$t_{\text{jry}} = \frac{s}{v} = \frac{s}{v_{\text{free}}(1 - \frac{k}{k_{\text{max}}})} \tag{4.3}$$

Based on (4.3), we use a matrix T_j to represent the node-to-node journey time for each road segment within the traffic network:

$$T_{\text{jry}} = \begin{array}{c c} & \begin{array}{cccc} n^1 & n^2 & \cdots & n^M \end{array} \\ \begin{array}{c} n^1 \\ n^2 \\ \vdots \\ n^M \end{array} & \begin{pmatrix} t^{1,1}(t) & t^{1,2}(t) & \cdots & t^{1,M}(t) \\ t^{2,1}(t) & t^{2,2}(t) & \cdots & t^{2,M}(t) \\ \vdots & \vdots & \ddots & \vdots \\ t^{M,1}(t) & t^{M,2}(t) & \cdots & t^{M,M}(t) \end{pmatrix} \end{array} \tag{4.4}$$

where M is the total number of network nodes. If there exists no link from node n^i to node n^j, then the journey time $t^{i,j}$ will be set to $+\infty$.

4.2.2 Event-triggered Mechanism

As previously mentioned, traffic event is defined as unexpected and sporadic scenarios that have severe impacts on traffic efficiency and safety. In this study, we primarily analyze the two typical scenarios along with their respective detection criteria:

- *Pedestrian gathering:* typically occurs at intersections with pedestrian crossings. As our traffic model is assumed to be non-signalized, vehicles should obey the “pedestrian-first” traffic rule. Hence, intersections experiencing pedestrian gatherings may lead to large street-crossing time and thereby cause congestion on the approach roads. In addition, since our users are CAVs operating in automated driving modes, the appearance of some kinds of pedestrian gathering poses serious latent risks, such as large numbers of children crossing the street during school commuting hours. To detect the occurrence of overcrowded gatherings, we introduce a pedestrian density threshold d_{thre} as the criteria

Algorithm 1 Event Trigger Algorithm

Input:

$G = (N, L)$ // Road network

$P^i = \{p_k, p_k.position \in n^i.area\}_{i=0}^{M}$ // Pedestrian set in each intersection area

$C^{i,j} = \{c_k, c_k.position \in l^{i,j}.area\}_{i,j=0}^{M}$ // Vehicle set on each road

$C^i = \{c_k, c_k.position \in n^i.area\}_{i=0}^{M}$ // Vehicle set in each intersection area

Output:

N_{eve} // Set of intersections with detected event

L_{eve} // Set of roads with detected event

Initialization: $N_{\text{eve}}, L_{\text{eve}} \leftarrow \emptyset$ **for** $n^i \in N$ **do**

  $d^i = P^i.amount/n^i.size$ **if** $d^i \geq d_{thre}$ ***or*** *there exists* $C^i.velocity \leq v_{thre}$ **then**

    $N_{\text{eve}} \leftarrow n^i$

  end

end

for $l^{i,j} \in L$ **do**

  if *there exists* $C^{i,j}.velocity \leq v_{thre}$ **then**

    $L_{\text{eve}} \leftarrow l^i$

  end

end

for assessment. When the measured density d^i in the vicinity of the intersection n^i is higher than d_{thre}, the intersection n^i will be regarded as overcrowded. The value of d_{thre} in practice is determined by various factors, such as city population, traffic type, etc.

- *Traffic accidents:* Traffic accidents may take place at any locale within the traffic network. The onset of such accidents often results in prolonged handling time, subsequently inducing congestion. Hence, vehicular speeds can be regarded as a criterion of the accident occurrence. Specifically, if a particular link $l^{i,j}$ or intersection n^i exhibits vehicles with speeds consistently below a threshold v_{thre}, it is inferred that the link $l^{i,j}$ or the intersection n^i has become a locus of the traffic accident.

In Algorithm 1, we show the traffic DT-based traffic monitoring process and its output will be used for route planning strategy. This process is permanently repeated to update the sets of nodes and links that are influenced by aforementioned two kinds of events.

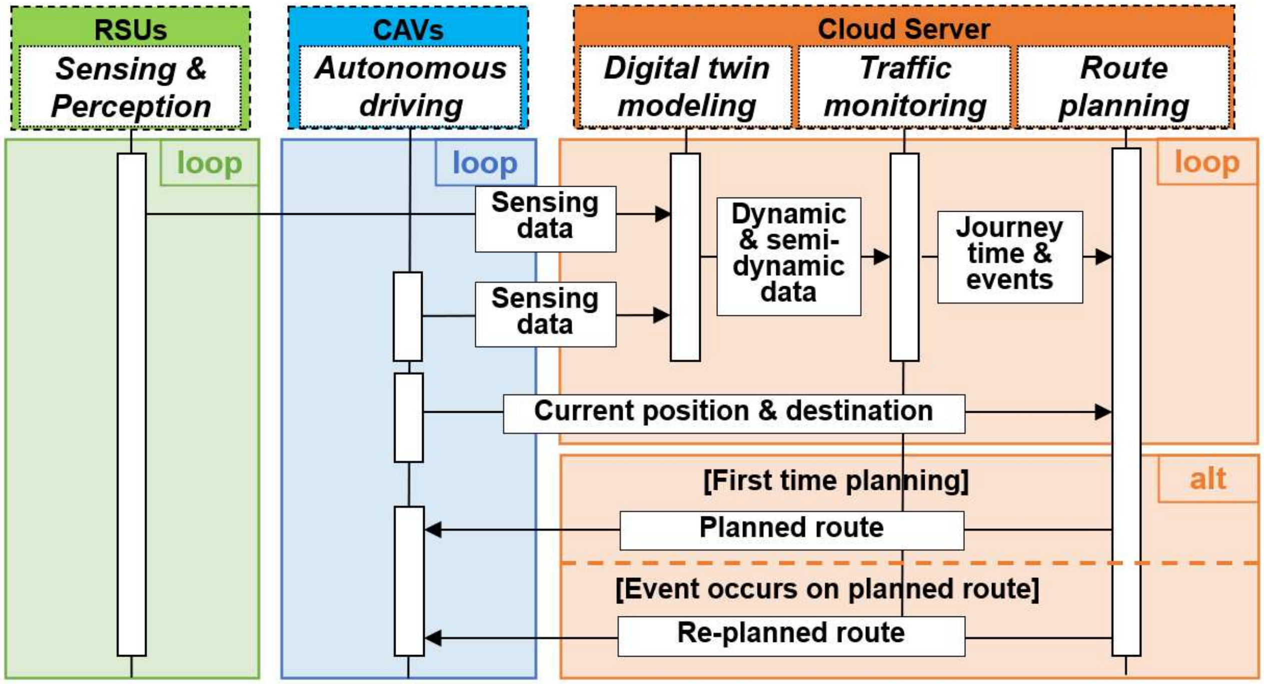


Figure 4.2: Flowchart of the SMDT-based CAV navigation system.

4.2.3 Workflow of CAV Navigation System

Based on the *journey time calculation* and *event-triggered mechanism* functionalities of SMDT, the solution of CAV navigation system can be designed as a cooperative event-triggered route planning process. The cooperative aspect is manifested by the necessity for all CAV users to upload their sensor data. This collaborative data sharing serves to effectively compensate for areas not covered by RSU sensors, ensuring that the cloud-based traffic digital twin provides a more accurate representation of the real-time traffic conditions in the road network and enhances monitoring of traffic events. On the other hand, the event-triggered mechanism works when traffic events are detected within the network. Such incidents then trigger route re-planning for users whose original paths overlap with these events.

Algorithm 2 Cooperative Event-triggered Route Planning

Input: $G = (N, L)$ // Road network

T_{jry} // Journey time of each road link

$U = \{u^i\}_{i=1}^{M_U}$ // Set of CAV users already in road network

$U_{\text{new}} = \{u_n^i\}_{i=1}^{M_N}$ // Set of new CAV users entering road network

$R(u_i)$ // Planned route of each user already in road network

N_{eve} // Set of intersections with detected event

L_{eve} // Set of roads with detected event

Output: $R_{\text{new}}(u_i)$ // Re-planned route for each user

Initialization: $R_{\text{new}} \leftarrow \emptyset$

```
for u_n^i ∈ U_new do
    n_start := u_n^i.CurrentPosition
    n_end := u_n^i.Destination
    R_new(u_n^i) := Dijkstra(N, L, T_jry, n_start, n_end)
end for
for n^i ∈ N_eve do
    for n^j ∈ N_eve \ {n^i} do
        if T_jry(n^j, n^i) ≠ +∞ then
            T_jry(n^j, n^i) := +∞
        end if
    end for
end for
for l^{i,j} ∈ L_eve do
    T_jry(n^i, n^j) := +∞
end for
for u^i ∈ U do
    for l^{i,j} ∈ R(u_i) do
        if there exists T_jry(n^j, n^i) = +∞ then
            n_start := u^i.CurrentPosition
            n_end := u^i.Destination
            R_new(u^i) := Dijkstra(N, L, T_jry, n_start, n_end)
        end if
    end for
end for
```

Figure 4.2 summarizes the functional blocks of the proposed CAV navigation system and illustrates the symbiotic relationship between RSUs, CAVs, and the cloud in the context of route planning strategy. RSUs contribute to the sensing and perception layer, capturing environmental data that is subsequently ingested by the cloud server. Concurrently, CAVs engage in autonomous driving, continuously collecting and transmitting their sensing data, together with current positions and intended destinations, to the cloud. On the cloud plan, a *traffic DT* can be developed using sensing data sourced from RSUs and CAVs, ensuring a real-time reflection of traffic conditions. This model aids in efficient traffic monitoring, capturing critical parameters such as journey time and the occurrence of events. The cloud then conducts route planning, where an initial path is determined for new entering CAVs, and can be re-routed should any unforeseen event happen on the pre-planned trajectory.

The pseudocode of the proposed route planning algorithm is shown in Algorithm 2. The procedure operates on the static road network G and the predicted journey T_{jry}, which is dynamically updated as the algorithm progresses to ensure real-time adaptation to changing traffic conditions. For the CAV users just entering the network, denoted as U_{new}, their initial route is generated using Dijkstra algorithm. Here, we note that the Dijkstra algorithm is employed to generate the fastest path, which is facilitated by assigning each road link's journey time as its weight, and the output is a set of links connecting a CAV user's current location to destination. If some events are detected at intersections (i.e., nodes), the journey time for links pointing towards these intersections will be set to infinity, to help users circumvent such events. Similarly, if events happen on specific road links, their journey time will also be set to infinity. Then the planned routes of CAV users already within the road network should be re-evaluated. If any part of a user's planned route includes a link with an infinite journey time (due to an event), the user's route is recalculated using Dijkstra algorithm. This ensures that all users are always provided with an optimal route considering the latest information about the network's conditions. This route planning algorithm dynamically adapts to real-time road conditions, ensuring optimal journey paths for users while mitigating the impact of unforeseen events. This process is permanently repeated to proactively respond to instantaneous changes in the network, offering users optimal journey time paths through event-triggered updates.

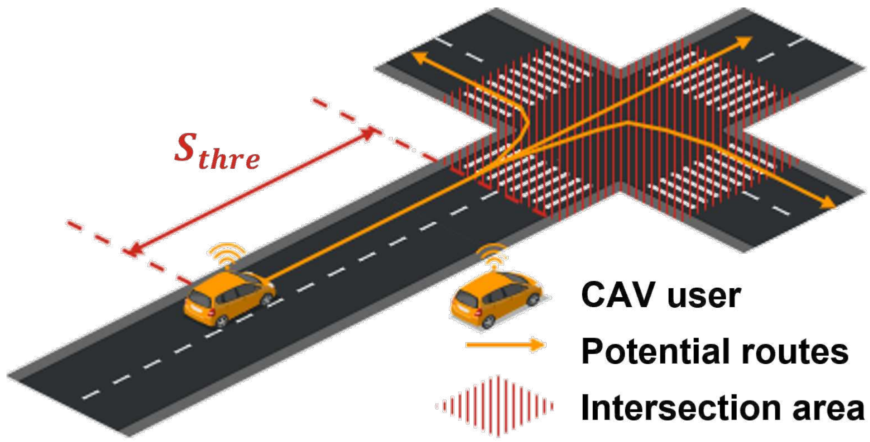


Figure 4.3: Illustration of route-planning request distance.

4.3 Requirements for SMDT-based CAV Navigation System

The SMDT-based CAV navigation system features a centralized architecture focusing on two core processes: *traffic DT modeling* and *route planning service*. To achieve *traffic DT modeling* over the cloud plane with RSU and CAV sensing data, sensor-equipped RSUs are adopted for collecting raw data nearby traffic intersections, and CAVs with onboard sensors driving within the road network serve to supplement traffic conditions beyond the sensing range of RSUs. As for *route planning service*, it is important to establish reliable communication to ensure the dissemination of planned routes. Upon near entering into the road network, the user should "tell" the cloud its impending presence, initiating continuous uploads of its position and desired destination, and then the cloud responds by furnishing an initially planned route. Since the initial planned route might make the user alter their predetermined direction at an intersection, as shown in Figure 4.3, the request should be initiated prior to the user entering the intersection. Here, we define a threshold distance S_{thre} at intersections, i.e., when the distance between the CAV user and the intersection area reaches S_{thre}, the user will send a route planning request to the cloud.

The threshold distance S_{thre} is determined with the consideration of comfort issues. The vehicle should have enough time to process the planned route before reaching the intersection. Given the presence of pedestrians crossing at intersections, it is necessary for the vehicle to comfortably decelerate and stop before reaching the intersection. According to the Institute of Transportation Engineers (ITE) recommendations, a comfortable deceleration rate a_{comfy} should be less than 10 ft/s^2, equivalent to 3.048 m/s^2 [165]. Hence, S_{thre} can be decided with the maximum speed (i.e., free-flow speed v_{free}) as:

$$S_{\text{thre}} = \frac{v_{\text{free}}^2}{2a_{\text{comfy}}} = 0.164\ v_{\text{free}}^2 \tag{4.5}$$

Communication links in the *traffic DT modeling* and *route planning service* processes correspond to two V2X use cases proposed by 3GPP [166], i.e., sensor and state map sharing (SSMS) and information sharing for high/full automated diving, respectively. SSMS enables the sharing of raw or processed sensor data to build collective situational awareness, which allows the central server to access and gather real-time traffic information obtained by RSU sensors. The information sharing use case encompasses two scenarios: cooperative perception and cooperative maneuver. Cooperative maneuver necessitates the sharing of detailed planned trajectories among all participating vehicles via V2X communication, which permits the central server to exert control over the vehicular routes.

To assess whether the proposed SMDT platform can meet the necessary requirements of the above two use cases, the respective key performance indicators (KPIs) are summarized in Table 4.1. The requirements in terms of reliability and E2E latency are defined in the same way as 3GPP technical report (TR) 22.886 [22]. By definition, reliability refers to the probability of successful transmission reliability and E2E latency refers to the one-way time delay it takes to deliver a piece of information from a source to a destination, without the application-layer processing delay. In this study, E2E latency of SSMS and information sharing use cases are defined respectively as the unidirectional communication delay between RSUs and the cloud, termed as T_{I2N}, and between the CAVs and the cloud, termed as T_{V2N}. In addition, we also define the total latency of *traffic DT modeling* and *route planning* pro-

Table 4.1: V2X Use Cases and Respective KPIs

Use cases	Reliability (%)	E2E latency (ms)	Total latency
SSMS	95	10	$T_{\text{dt}} \ll T_{\text{svc}}$
Information Sharing	-	100	$T_{\text{svc}} < 0.164\ v_{\text{free}}$

cesses. In our system, the basis for decision-making is the dynamic traffic DT, which must represent real-time traffic conditions. Consequently, the latency in DT modeling T_{dt} needs to be significantly shorter than the latency in the route planning service process T_{svc}, to ensure that the traffic DT used for decision-making can be regarded as real-time. The DT modeling latency T_{dt} mainly consists of RSU edge computational time and one-way E2E latency from LiDAR to cloud server, expressed as:

$$T_{\mathrm{dt}} = T_{\mathrm{RSU}} + T_{\mathrm{I2N}} \tag{4.6}$$

Considering the route planning service is based on a request-response mechanism, it is imperative to ensure that users can receive, load, and execute the new planned route before entering the intersection. The route planning latency T_{svc}, spanning from the moment the user reaches the request distance S_{thre} to the execution of the route, should meet:

$$T_{\mathrm{svc}} \leq \frac{S_{\mathrm{thre}}}{v_{\mathrm{free}}} = 0.164\ v_{\mathrm{free}} \tag{4.7}$$

where the latency of offering route planning service T_{svc} includes time delays caused by localization T_{local}, route loading and execution T_{exe} in the autonomous driving system, cloud computing T_{cloud}, and bidirectional V2N communication T_{V2N}.

$$T_{\mathrm{svc}} = T_{\mathrm{local}} + T_{\mathrm{exe}} + T_{\mathrm{cloud}} + 2\ T_{\mathrm{V2N}} \tag{4.8}$$

4.4 Proof-of-Concept Demonstration of CAV Navigation System

In this section, we present an outdoor PoC experiment of SMDT-based route planning for autonomous driving. The experimental setup for route planning process in the Sci Tokyo Smart Mobility Field is first explained. Then we study two practical cases that effectively reflect key functions of proposed system.

4.4.1 PoC Setup in Science Tokyo Smart Mobility Field

Given the context of our Smart Mobility Field located within a campus environment, it inherently experiences minimal vehicular flow. This means that under normal conditions,

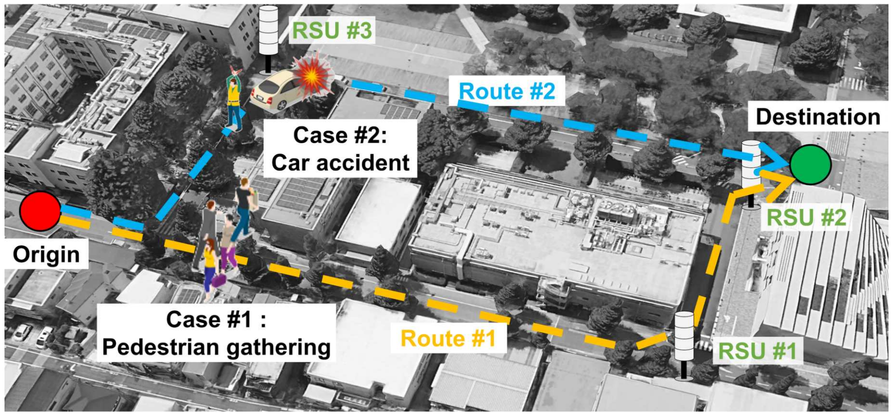


Figure 4.4: An overview of experiment setup.

vehicle volume is not a determinant of journey times. However, the driving environment is relatively complex due to the frequent presence of pedestrians and cyclists on the roadways. Coupled with a simple road network structure, our PoC experiment does not emphasize the *journey time calculation* function or the usage of the Dijkstra algorithm to search for a route for autonomous vehicles. Instead, the focus is on demonstrating the *event-triggered mechanism*, i.e., utilizing the established *traffic DT* to detect and identify traffic events, then subsequently providing route planning for the CAV to bypass such incidents.

In this experiment, the ego vehicle corresponds to CAV #1 as shown in Figure 3.2(b). Its origin and destination are depicted in Figure 4.4. There are two routes with approximately equal distances bridging the start and end points: Route #1 (206 m) and Route #2 (210 m),

Table 4.2: Roles of CAVs, RSUs, and the Cloud in PoC Experiment.

Component	Functionalities
RSU	1. Detect and track objects within sensing range; 2. Detect traffic accidents within sensing range.
CAV	1. Operate in a fully autonomous driving mode; 2. Detect and track objects within sensing range; 3. Detect traffic accidents within sensing range.
Cloud server	1. Collect traffic information from RSUs and CAVs; 2. Model and visualize DT; 3. Assign routes for CAVs to avoid accidents.

where Route #1 is regarded as a default route. Under normal conditions without any traffic events, the cloud server would select Route #1 and send its waypoint file to CAV users by default. In Case #1, A *pedestrian gathering* event occurs on the default Route #1, which is not within the sensing range of any RSU. Both CAV #1 (i.e., ego vehicle) and CAV #2 drive through the road network sequentially from origin to destination, where the maximum speed of CAVs $v_{\max}$ are set to match the road speed limit, i.e., 20 km/s in the campus. As CAV #2 first enters the network and travels along the default Route #1, it will encounter the *pedestrian gathering* and stop in front of it at a safe distance. This event will be thus detected by CAV #2 sensor. In Case #2, we park CAV #2 on Route #2 to simulate the occurrence of a *traffic accident*, positioned within the sensing range of RSU #3. Then this event can be detected by RSU #3. Some main responsibilities of RSUs, CAVs, and the cloud server in the PoC experiment are summarized in Table 4.2.

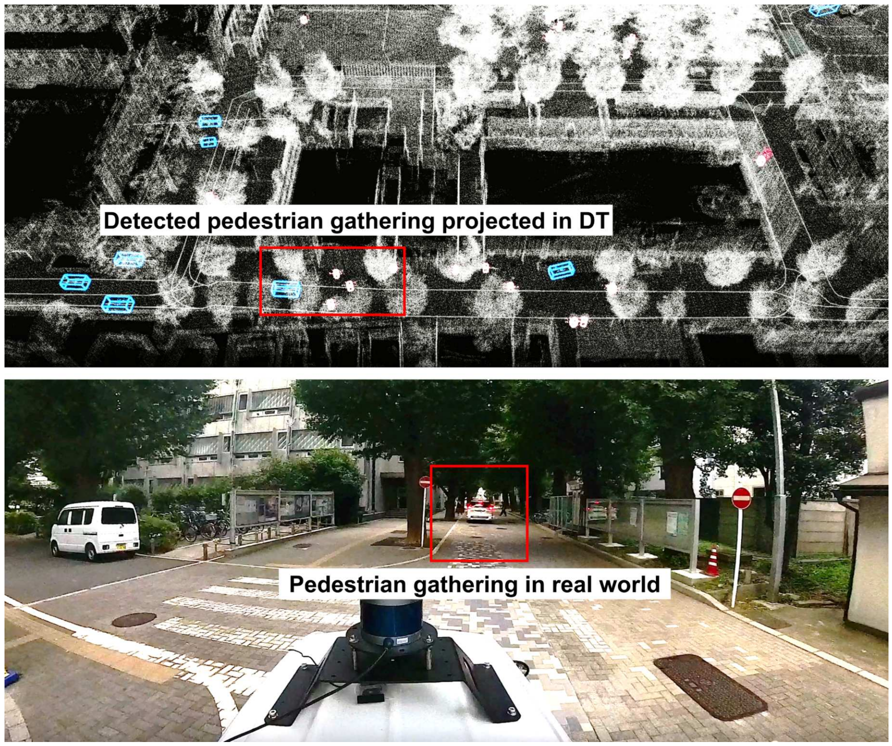
4.4.2 Case Study: Traffic DT-based Route Planning

According to these two cases, a PoC experiment is conducted and a demonstration video[1] is available online, showcasing the practical implementation and effectiveness of the SMDT-based CAV navigation system in real-world scenarios.

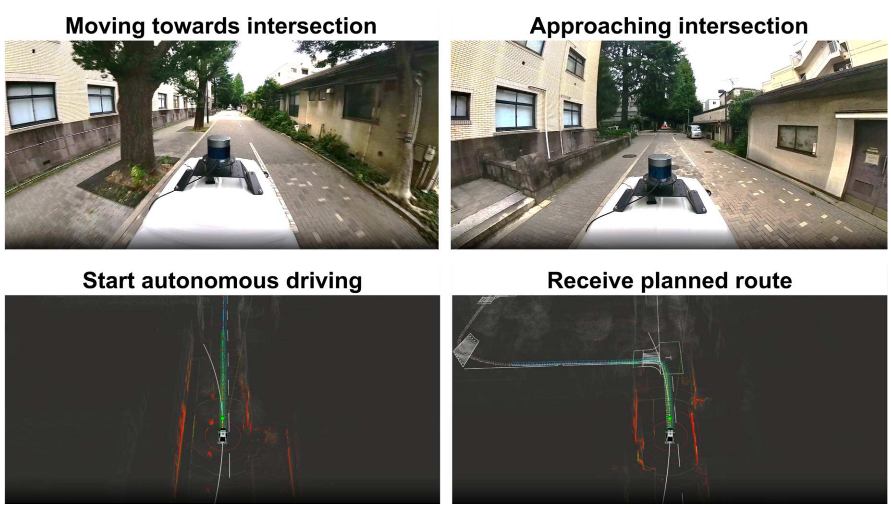
4.4.2.1 Case #1 Pedestrian gathering detected by CAV

Figure 4.5 shows the results of SMDT-based route planning process in Case #1, Figure 4.5(a) gives a view of traffic DT on cloud server, where the pink bounding boxes represent detected pedestrians. It can be seen that there are three pedestrians walking on the roadway of Route #1. This information is then captured by CAV #2, which is the first CAV entering the road network, and subsequently uploaded to the cloud. A car-following camera mounted on CAV #1 to record its driving process, also captures this scene, where CAV #2 halts in front of the gathering crowd. Figure 4.5(b) shows the autonomous driving process of CAV #1 from the perspectives of the car-following camera and the Autoware visualization tool Rviz, respectively. During the initialization phase, CAV #1 is proceeding straight, poised to enter the road network. When CAV #1 arrives at the predetermined threshold distance S_{thre} from the intersection area, it will start sending its position to the cloud through V2N communication to notify its impending entry into the network and requests route planning

[1]https://youtu.be/3waQwlaHQkk



(a) Visualization of traffic DT and real-world image showing the congestion caused by pedestrian gathering.



(b) Car-following camera view and ego vehicle Rviz view showing the executed route.

Figure 4.5: Illustration of traffic DT and autonomous driving operation results in Case #1.

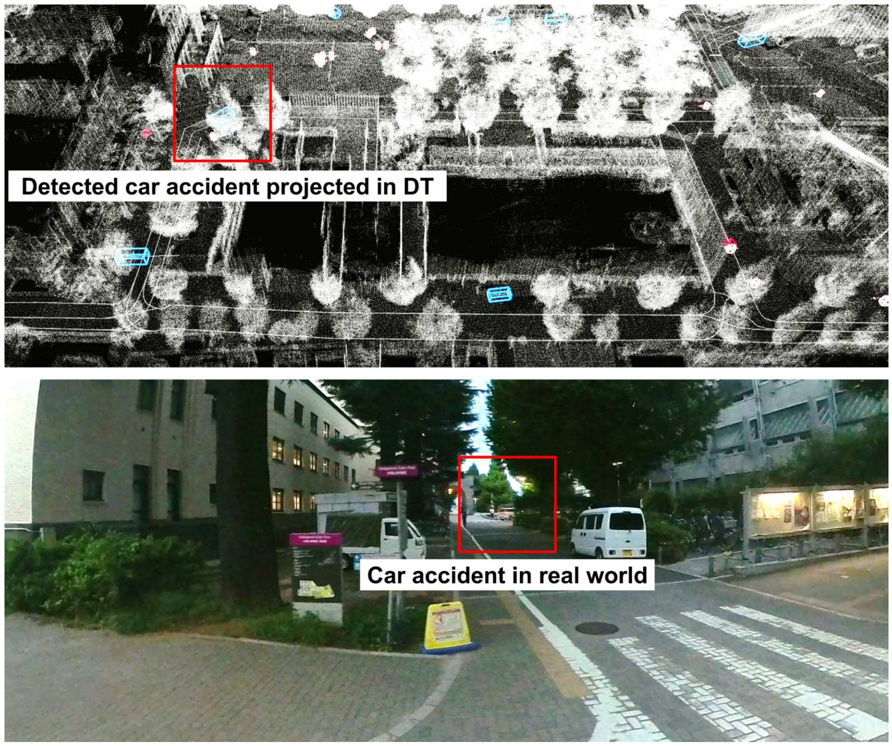
service. Then the cloud sends the Route #2 's Waypoints file to CAV #1 to help it avoid pedestrian gathering area and navigate it to destination. Consequently, CAV #1 executes the planned route to turn left at the intersection and finally arrives at the destination. In this experiment, the threshold distance S_{thre} is determined according to (4.5) with maximum vehicle speed:

$$S_{\text{thre}} = 0.164\ v_{\max}^2 = 5.062\ m \tag{4.9}$$

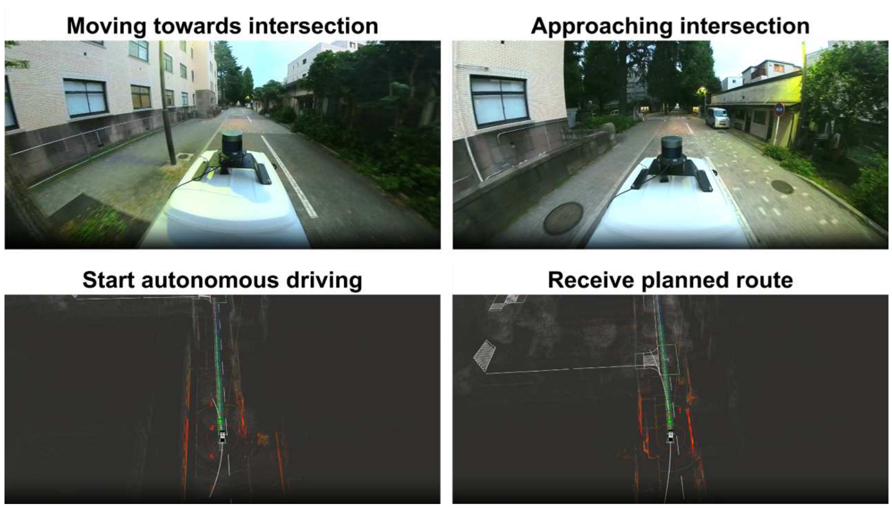
4.4.2.2 Case #2 Traffic accident detected by RSU

Similar to Case #1, Figure 4.6(a) shows the traffic DT, where a parked car can be found on Route #2, represented by a blue bounding box. Under such circumstances, the cloud system would deduce that a car accident has occurred on Route #2. In Figure 4.6(b), we also give a real-world image from car-following camera on CAV #1 that records the parking car and executed route in Rviz. In this case, the cloud sends Route #1 's Waypoints to CAV #1 to help it avoid the car accident on Route #2.

The case study demonstrates the effectiveness of SMDT-based route planning for autonomous vehicles: *(i)* the system's ability to detect and respond to real-time events, such as pedestrian gatherings and traffic accidents, and *(ii)* reliable route planning service to achieve the control on vehicle level. This underscores the system's potential for real-world application in dynamic traffic environments. However, challenges for future development mainly include safety and robustness issues. Safety concerns primarily arise from the reliance on the V2X communication network and the accuracy of traffic monitoring, especially in unpredictable urban environments with high CAV density. Robustness is challenged by varying environmental conditions and sensor limitations. To address these, future work could: *(i)* enhance V2X network resilience by implementing advanced encryption and cybersecurity protocols and deploying decentralized communication networks, such as blockchain technology, *(ii)* expand the variety of sensors and integrate sensor fusion techniques to provide a more comprehensive and fail-safe approach to environmental perception, *(iii)* explore the use of big traffic data analytics and deploy advanced machine learning algorithms to analyze vast amounts of traffic data more efficiently and accurately, and *(iv)* develop more sophisticated decision-making models that consider a wider range of variables, such as road surface and weather conditions.



(a) Visualization of traffic DT and real-world image showing the congestion caused by pedestrian gathering.



(b) Car-following camera view and ego vehicle Rviz view showing the executed route.

Figure 4.6: Illustration of traffic DT and autonomous driving operation results in Case #2.

4.5 Large-Scale Traffic Simulation and System Evaluation

To evaluate the effectiveness of the proposed CAV navigation system and the crucial KPIs of the implemented SMDT platform, a set of simulations and experiments are conducted. SUMO is utilized as a large-scale traffic simulator, where we can import test areas from the OpenStreetMap (OSM), then generate random vehicles and control them with TraCI4Matlab [167]. TraCI4Matlab acts as a bridge between MATLAB and SUMO, enabling real-time traffic flow management and dynamic control of individual vehicles. Using this interface, we can generate traffic flows by specifying vehicle insertion rates, routes, and types, creating a variety of traffic conditions for testing. Additionally, TraCI4Matlab allows for the simulation of traffic accidents by introducing specific disruptions, such as temporary lane closures, reduced vehicle speeds, or random vehicle stops, mimicking real-world traffic events. Subsequently, we deploy the cooperative event-triggered route planning algorithm in MATLAB to realize real-time path planning for CAV users. Finally, we empirically measure several metrics to validate whether our implemented platform meets the requirements posed by the CAV navigation system.

Table 4.3: Simulation Parameters

Parameters	Description	Value
ΔT	Sampling time	1 s
T_{sim}	Total simulation time	600 s
M	Amount of nodes	90
M_{link}	Amount of links	504
N_{vel}	Amount of vehicles	300
N_{user}	Amount of CAV users	10 - 100
P_{user}	Proportion of CAV users	3.3 - 33.3%
E	Amount of traffic events	0 - 10

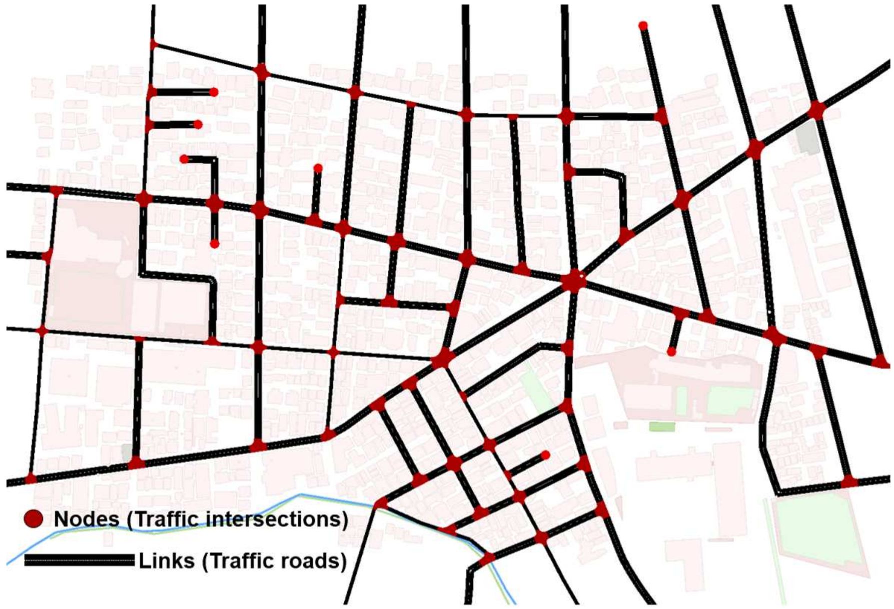
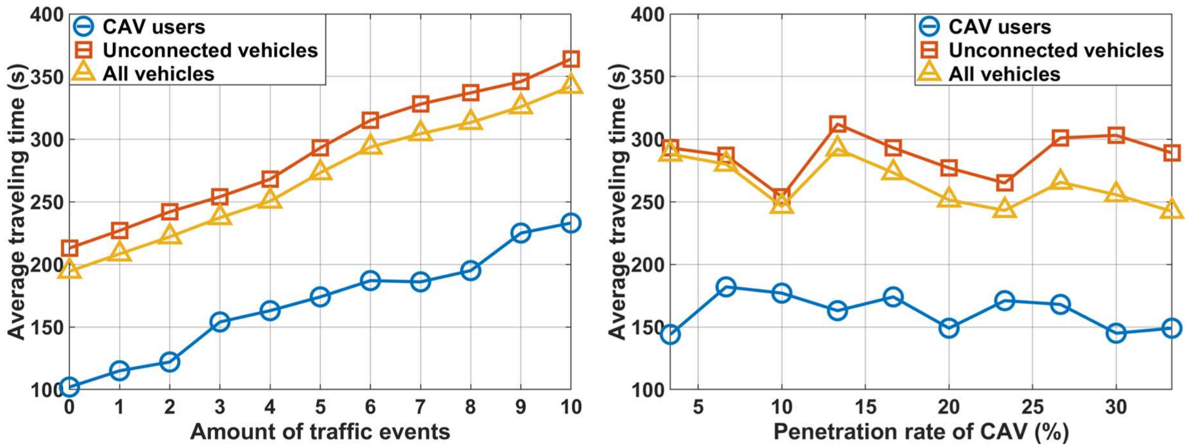


Figure 4.7: Traffic network of SUMO simulation in Tokyo, Japan.

4.5.1 Efficiency and Safety Improved by CAV Navigation System

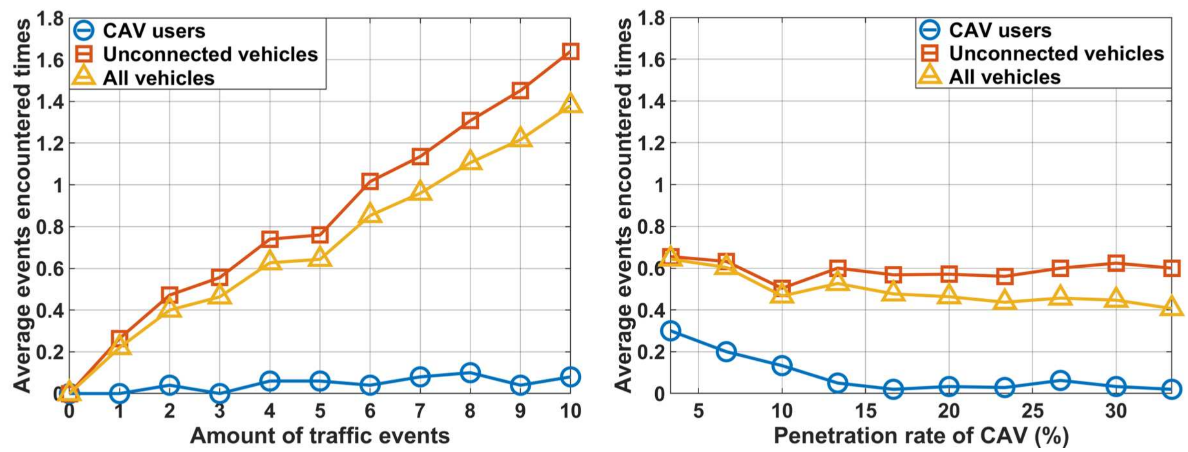
In SUMO-based traffic simulation, we import an urban traffic area located in Tokyo as illustrated in Figure 4.7. The speed limit for each road remains consistent with the default values in OSM, and every intersection is designed to be non-signalized. During the real-time simulation process, at each sampling time, a random number of vehicles enter the traffic network with both their origins and destinations being randomized. A certain proportion of these vehicles are designated as CAV users, for whom we provide real-time route planning services. In contrast, the remaining vehicles are classified as unconnected vehicles. Lacking dynamic traffic information, they opt for the shortest path between their origin-destination (OD) pairs. Throughout the simulation, a certain number of traffic events, i.e., *pedestrian gathering* and *traffic accidents*, occur at random places in the traffic network. Table 4.3 summarizes the main parameters used in this simulation.

Figure 4.8 shows the improved traffic efficiency and road safety with our CAV navigation system. We employ the frequency of encounters with traffic events as an evaluation metric for road safety since such extreme events could pose latent threats and safety concerns for vehicles passing by. In Figure 4.8(a), As the number of events increases, both CAV users and unconnected vehicles experience longer driving time to traverse the network. This is predominantly due to that congestion caused by traffic events might propagate to other road



(a) Average travel time with increasing penetration rate of CAV P_{user} ($E = 5$).

(b) Average travel time with increasing number of traffic events E ($P\text{user} = 16.7\%$).



(c) Average event encountered times with increasing penetration rate of CAV P_{user} ($E = 5$).

(d) Average event encountered times with increasing number of traffic events E ($P_{\text{user}} = 16.7\%$).

Figure 4.8: Simulation results of traffic efficiency and road safety.

segments. Nevertheless, the travel time for CAV users is significantly smaller than that of unconnected vehicles. In Figure 4.8(b), it can be observed that, given a constant number of events, with the number of CAV users increasing, the efficiency of CAV users cannot be improved. However, the travel time for the overall traffic consistently shows a declining trend, where some fluctuations are attributed to the random selection of OD pairs and the random occurrence locations of traffic events. As can be discerned from Figure 4.8(c), with the occurrence of an increasing number of events, it can be seen that the frequency with which CAV users encounter these events does not show an upward trend. This suggests that the *event-triggered mechanism* in our route planning strategy is effective in assisting users to circumvent such events. As depicted in Figure 4.8(d), with an increasing penetration rate of CAV users, the safety of both CAV users and the overall traffic can be improved, manifesting as a reduction in the frequency of encounters with traffic events.

4.5.2 System Evaluation Considering Communication Issues

In DT modeling, the packet delivery rate (PDR) for communication from RSUs to the cloud server is very critical and important, since significant packet loss may lead to incorrect planning decisions. When only processed sensing data (i.e., detection and tracking results) are transmitted, the PDR can approach 100%. However, when simultaneously uploading other levels of data, such as raw LiDAR point clouds, the PDR slightly decreases to approximately 99.53%, which still satisfies the reliability criterion for SSMS, which mandates a PDR greater than 95%. As for the process of providing route planning service, V2N communication is built upon the hypertext transfer protocol (HTTP). Since HTTP operates atop the transmission control protocol (TCP) at the application layer and TCP ensures data delivery through error detection and retransmissions, the reliability of data transfer under the HTTP protocol can be ensured in stable network conditions.

Some crucial communication and computational latency existing in our system are also measured, as summarized in Table 4.4. It can be found that both the maximum one-way I2N (1.74 ms) and V2N communication latency (42.1 ms) can meet the requirements proposed by 3GPP, which are 82.6% below the threshold of the max E2E latency for SSMS (less than 10 ms), and 57.9% below the threshold for information sharing (less than 100 ms).

The total latency for *traffic DT modeling* T_{dt} is primarily constituted by the latency derived from edge computing T_{dec} and the one-way I2N communication T_{I2N}, both of which are measured through experiments. The experimental setup for latency measurement in *traffic DT modeling* includes NVIDIA Jetson system equipped with an ARM Cortex-A78AE CPU (12 cores), 32 GB of memory, and Ubuntu 20.04 operating system. Each measurement was conducted over a duration of 10 minutes. Given that the quantity of objects within the LiDAR's sensing range has a large impact on the detection duration, we have conducted measurements of the latency spanning the initiation of the LiDAR to the computation of the detection outcomes many times at various intervals throughout the day. The edge computing exhibited a maximal latency of 153.41 ms, whereas the overall latency for the *traffic DT modeling* reached a peak of 155.15 ms, according to (4.6).

As for the *route planning service* latency, we separately measure V2N communication delay T_{V2N}, cloud computing time T_{cloud} for both traffic monitoring and route planning, and edge computing time for NDT-based localization and route loading/execution. The experimental setup for latency measurement in *route planning service* includes the cloud server equipped with an Intel Core i9 CPU, 64 GB of memory, and Ubuntu 22.04 operating system. Each measurement is repeated 10 times. Only the cloud computational time is measured with SUMO-based large-scale simulation, instead of with PoC experiment, because cloud computing in PoC just selects a route from two options, without searching for the fastest route. Based on (4.8), the maximal total latency of the *route planning service* reaches 810.59 ms, which complies with the requirement of CAV user receiving and executing the routing command prior to entering the intersection area. Additionally, the latency of *traffic DT modeling* is also significantly lower than that of *route planning service.*

In the real-world experiment, the frequency for LiDAR-based object detection is set at

Table 4.4: Communication Requirements and Performances

	Max. (ms)	Min. (ms)	Mean (ms)
DT modeling process			
I2N comm.	1.74	1.10	1.37
Edge comp. (detection)	153.41	70.01	106.23
T_{dt} Total Max.: 155.15 ms			
Route planning service			
V2N comm.	42.13	20.16	32.30
Cloud comp. (traffic monitoring)	56.29	42.72	45.07
Cloud comp. (route planning)	201.07	173.27	183.68
Edge comp. (NDT localization)	10.13	2.56	6.14
Edge comp. (route loading)	500.97	501.35	501.18
T_{svc} Total Max.: 810.59$<$ 0.164 $v_{\max} = 911.11$ ms			

30 Hz, which is regarded as the frequency of *traffic DT modeling*. Route planning, on the other hand, is executed on-demand, triggered when the CAV approaches an intersection within the threshold distance S_{thre}. Considering the update frequency and its associated delays, in extreme cases, the latency for *traffic DT modeling* will reach up to 188.48 ms, which is also much lower than that of *route planning service*. For simulations, the cloud computing capabilities operate with a sampling time of 1 s, which is larger than the measured maximum latency of *route planning service*. This setup ensures that cloud services can be completed in real time within each sampling period, demonstrating the cloud's ability to process critical information promptly and efficiently. Taking into account the three crucial aspects of reliability, latency, and frequency, the traffic DT can be regarded as real-time and effective in the context of CAV navigation system.

4.6 Summary

This Chapter presents the design and implementation of a CAV navigation system powered by the SMDT platform, addressing critical limitations in existing autonomous vehicle navigation frameworks. By integrating dynamic traffic information through V2X communication, the system provides real-time adaptive route planning and ensures efficient navigation even in the presence of unexpected events, such as accidents or pedestrian gatherings.

The Chapter demonstrates how the SMDT platform's hybrid architecture of cloud and edge computing supports seamless traffic monitoring and decision-making. PoC experiments validate the system's ability to detect traffic events and re-route vehicles effectively, underscoring its potential for real-world application. Simulation results further highlight the system's capacity to improve traffic efficiency and safety under varying conditions. In summary, the SMDT-based navigation system bridges the gap between static map-based approaches and dynamic traffic-aware frameworks, offering a scalable and robust solution for next-generation autonomous driving in urban environments.

Chapter 5

Local Digital Twin-Assisted Hybrid Autonomous Driving Systems

This Chapter proposes a novel local digital twin (LDT)-assisted hybrid autonomous driving system, specifically designed for traffic intersections to improve road safety and driving efficiency. First, in Section 5.1, we outline the motivations for developing LDTs, emphasizing their role in augmenting vehicle-road collaboration for safer and more efficient intersection management. In Section 5.2, we present the architecture of the LDT-assisted hybrid autonomous driving system, detailing the roles of roadside units (RSUs), connected and automated vehicles (CAVs), and the central cloud in achieving localized decision-making and real-time control. In Section 5.3, we describe the offline reinforcement learning (RL) framework used to extract human driving knowledge and generate intersection-specific driving agents, highlighting its integration with the proposed system. In Section 5.4, we explain the hardware deployment and software implementation of the system in a real-world testing environment, providing technical details about RSUs, CAVs, and communication infrastructures. Finally, in Section 5.5, we present the results of proof-of-concept (PoC) experiments and hardware-in-the-loop (HiL) simulations, showcasing the system's improvements in safety, efficiency, and real-time responsiveness at intersections. Finally, we summarize this Chapter in Section 5.6.

5.1 Motivations

The emergence of autonomous driving systems (ADS) demonstrates critical challenges that hinder ensuring road safety measures. This is primarily due to the necessity of processing substantial volumes of real-time data, including sensing, perception, localization, and decision-making, within stringent latency constraints [168]. Processing such data should occur within milliseconds to ensure that vehicles can react to dynamic driving environments in real time. Any delays in communications and computing could lead to incorrect or delayed decisions, which are especially dangerous in fast-evolving situations like intersections. To overcome such bottlenecks in individual autonomous vehicle systems, vehicle-road collaboration-based augmented intelligence has attracted significant interest recently (e.g., see [169] and [170]). For instance, such augmented intelligence can facilitate information exchange between CAVs and surrounding environments while also enabling the extraction of human driving expertise from dynamic and unpredictable traffic scenarios to enhance onboard intelligence. In particular, intersections in urban areas are the most hazardous as they consider the frequent presence of blind spots, vulnerable pedestrians, and stop-and-go behaviors [171]. A report from the US Federal Highway Administration (FHA) reveals that intersections are hotspots for vehicular accidents, accounting for 40% of all traffic crashes and 50% of serious collisions occurring exclusively within these areas [172]. Hence, the development of intersection-specific augmented intelligence becomes an indispensable solution to elevate the levels of intelligence in CAVs, whereby this can contribute to enhanced levels of safety measures.

With recent advances in digital twin (DT) technologies, the concept of mobility digital twins (DTs) serves as an exemplary *embodiment of augmented intelligence*, offering a robust platform to address various on-road challenges [150,153,148]. Mobility DTs provide a multi-layer framework that integrates real-time data acquisition, simulation, and predictive modeling to mirror the behaviors of physical systems within a virtual environment. In the context of autonomous driving, Mobility DTs can capture the intricacies of traffic dynamics, road infrastructure, and vehicle behaviors to create a comprehensive model that continuously adapts to real-world conditions [80]. On the one hand, Mobility DTs encompass comprehensive global traffic information, thereby providing connected users with far-reaching, computation-intensive services such as global vehicle routing that extend well beyond immediate sensory capabilities [173], which enables autonomous driving systems to make smarter decisions based on system-level insights derived from real-time global traffic data. On the other hand, MDT incorporates distributed intelligence across various smart entities [42], including CAVs, smart traffic signals, and RSUs. This distributed framework facilitates the provision of localized, delay-sensitive services. By leveraging real-time data and computational power within these interconnected smart entities, Mobility DTs can dynamically adapt to the specific conditions of different traffic areas, ensuring more responsive and effective decision-making for CAVs.

Therefore, we introduce a novel concept named "local DT (LDT)", a specialized DT system designed to monitor, analyze, and optimize traffic behaviors within specific areas, i.e., traffic intersections. Acting as a subsystem of a global DT, the LDT enhances flexibility and reusability by leveraging local services, storage, models, and edge connections to ensure faster response [174]. In contrast to global DT, which focuses on large-scale traffic patterns, LDTs are suitable to deliver low-latency responses by processing data locally at the edge, enabling real-time decision-making for time-sensitive tasks such as vehicle maneuvering, intersection management, and adaptive traffic signal timing (e.g., see [68] and [69]). In addition, this localized approach allows the system to address the intricate and dynamic nature of intersections more effectively than a generalized global system. Considering the unique characteristics of each traffic intersection, including its road topology, the volume of traffic, and the diversity of traffic participants, LDTs can generate strategies tailored to each intersection scenario. The overarching goal of this paper is to propose an LDT-assisted hybrid vehicle maneuvering system in intersection areas that explores the potential revolution of autonomous driving systems brought by vehicle-road collaboration-based augmented intelligence. Through this system, we aim to enhance the decision-making capabilities of CAVs with LDT-based cooperative perception and local driving strategies provided by RSUs, enabling them to navigate vehicles through intersections more safely and efficiently.

5.2 Design of Hybrid Autonomous Driving System

In this section, we introduce the architecture of our LDT-assisted hybrid autonomous driving system, leveraging computational resources at the central cloud and edge (RSUs and CAVs), as shown in Figure 5.1. The technical requirements of the proposed system are discussed based on the 3GPP standards for sensor and state map sharing (SSMS) and artificial intelligence (AI)/machine learning (ML)-based automotive networked systems.

5.2.1 Overall System Design

The proposed LDT-assisted hybrid autonomous driving system aims to enhance road safety and the driving efficiency of autonomous vehicles at intersections, which are fraught with risks due to frequent pedestrian crossings and blind spots, posing challenges for autonomous vehicles. In our system, we assume that RSUs are deployed in complex traffic environments,

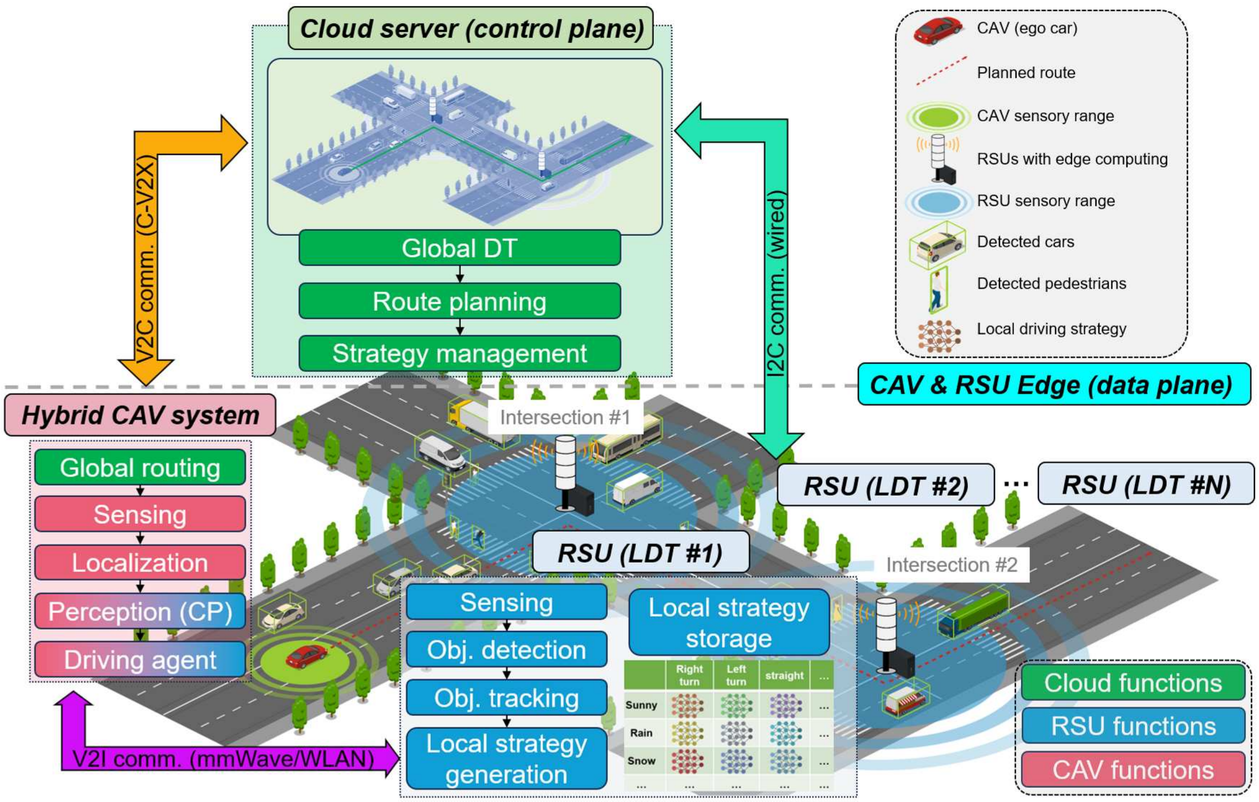


Figure 5.1: LDT-based hybrid autonomous driving system architecture.

including non-signalized intersections. They are equipped with various sensors and computing units. In addition, we assume that the CAVs (i.e., target users) in the proposed system have achieved high-level automation with all necessary functional blocks, including sensing, localization, environmental perception, prediction, planning, and motion control. A computationally capable central cloud is also deployed at the top of this system. The detailed functionalities of RSUs, CAVs, and the central cloud are discussed next:

1) RSU edge: The RSUs persistently monitor the traffic environment using mounted sensors and advanced objection detection techniques. In this system, RSUs are the hosts of LDTs. They perform three critical tasks: *(i)* upload real-time processed perception information to the cloud; *(ii)* share real-time perception information with surrounding CAVs; *(iii)* record, analyze, and extract knowledge from the driving behaviors of passing-by vehicles, then generate *local driving strategies* tailored for locational intersections within specific environmental conditions, such as varying times of day and weather patterns.

2) Central cloud: The central cloud establishes a real-time GDT, after receiving and synchronizing perception data from distributed RSUs, based on which the central cloud also plans global routes for each CAV user in the traffic network. Since the cloud is aware of all the intersections that the CAVs will pass through, it can determine and "inform" them of the most appropriate *local driving strategies* to guide CAVs effectively by combining real-time assessments of current conditions.

3) CAV edge: The CAVs follow the global routes planned by the central cloud. When CAVs approach the intersections managed by RSUs and enter their vehicle-to-infrastructure (V2I) coverage, the CAVs start downloading *local driving strategies* from RSUs and follow. Instead, they depend on their inherent autonomous driving systems while driving on road segments without RSUs. The operations of CAV systems at RSU-assisted intersections can be separated into three steps: *(i)* request *local driving strategies* from RSUs; *(ii)* receive real-time perception data from RSUs and fuse with onboard sensing; *(iii)* switch the driving agent from inherent motion planning and motion control modules to *local driving strategies*. An intersection-specific hybrid autonomous driving system is thus achieved.

For example, in Figure 5.1, a CAV scheduled to traverse the intersections #1 and #2 will request the *local driving strategies* that match current time and weather from RSU #1 and #2, respectively. Because such *local driving strategies* are extracted and learned from local driving behaviors, adopting them allows CAVs to make more reasonable decisions tailored for corresponding local areas. Upon the CAV entering the V2I communications coverage of RSU #1 and #2, the cooperative perception service is provided to extend the horizon of onboard sensing. Such enhanced perception endows the CAV with a comprehensive understanding of the traffic environment and the capability to identify potential hazards concealed within blind spots.

This system design capitalizes on heterogeneous communications and computing resources on edge and cloud. The edge computing on RSUs processes raw sensor data into object-level detection information, in order to reduce the large transmission overhead and delays to the CAVs. Since vehicle control is an extremely delay-sensitive and safety-critical task, the RSUs will directly transmit their *local driving strategies* through V2I, to be executed by edge computing on CAVs. In the system, *local driving strategies* dissemination is managed by the control plane, which is located in the central cloud, while the data plane is at the edge. The main reasons are that: *(i)* the cloud system is responsible for planning the global route for CAVs. As a result, the cloud can prefetch the required *local driving strategies* along the planned route and delegate RSU edges to transmit them in advance, and *(ii)* considering the large data size of AI/ML-based models, V2I transmission, compared to V2N transmission, is much more cost-efficient and will not lead to the model reception and execution failure on CAVs due to unstable V2N delay.

5.2.2 Requirements Analysis Considering Communications Issues

The proposed system aims to enhance the safety and efficiency of CAVs by utilizing V2I communications to facilitate the download of *local driving strategies* and the sharing of perception data. To achieve such functions, the system must meet certain communications requirements to ensure reliable and timely data exchange. Therefore, it is necessary to define specific key performance indicators (KPIs) that the system should fulfill. *Local driving strategies* downloading and perception data sharing correspond to an AI/ML model transfer use case and a V2X use case proposed by 3GPP [166,175], called AI/ML based automotive networked systems and SSMS, respectively. By definition, AI/ML-based automotive networked systems frequently exchange and share AI/ML model layers in a distributed and/or federated network as determined by the system in response to a change of events, conditions, or emergency situations to improve system accuracy. SSMS enables the sharing of raw or processed sensor data to build collective situational awareness, thereby enabling safety-critical applications like intelligent intersections to ensure the safety of all road users, including pedestrians and

Table 5.1: Communications KPIs of AI/ML based automotive networked system and SSMS use cases

Use cases	**AI/ML based automotive networked system**	**Sensor and state map sharing (SSMS)**
Reliability (%)	-	90
E2E latency (ms)	500∼1000	10
Peak data rate (Mbps)	100	25
Data size (MB)	2∼40	-

emergency vehicles.

To assess whether the proposed hybrid autonomous driving system can meet the necessary requirements of the above two use cases, some key performance indicators (KPIs) that need to be validated are summarized in Table 5.1. For the AI/ML-based automotive networked system use case, some required communications KPIs are determined based on 3GPP technical report (TR) 22.874 [142], including data size, end-to-end (E2E) latency, and peak data rate for data downloading. In the proposed system, CAVs shall download a series of AI models of *local driving strategies* for different intersections from RSUs. Here, we define data downloading as that CAV downloads only one of the AI models. The proposed system should be able to support downloading of data with a maximum size of 2$\sim$40 MB with a latency of up to 500$\sim$1000 ms and a peak data rate of up to 100 Mbps.

According to 3GPP TR 22.886 [22], the requirements of SSMS include reliability, E2E latency, and peak data rate. The reliability refers to the probability of successful transmission, which should be higher than 90%. The E2E latency refers to the one-way delay for delivering a piece of information from a source to a destination, without the application-layer processing delay. In this system, the E2E latency of SSMS is defined as the one-way communications delay of cooperative perception, where RSU shares real-time perception information with surrounding CAVs, and the latency should be less than 10 ms. The peak data rate should be higher than 25 Mbps.

5.3 Approach: Offline Reinforcement Learning Framework for Localized Learning

In this section, we propose an offline RL framework to extract and learn human driving knowledge. Offline RL is commonly applied to autonomous driving systems because it breaks the constraint that agents should interact with the environment, thereby avoiding risky exploration in real driving processes [176,177]. Unlike online RL, which requires continuous interaction between agents and the environment, offline RL allows us to safely utilize vast amounts of pre-collected driving data from human-driven vehicles to train driving models without exposing the vehicles to hazardous real-world conditions. Moreover, compared to non-RL approaches, which may struggle to generalize across diverse driving conditions and require extensive hand-crafted rules, offline RL offers a more flexible and data-driven way to learn complex driving behaviors. In our system, the role of gaining human knowledge is placed

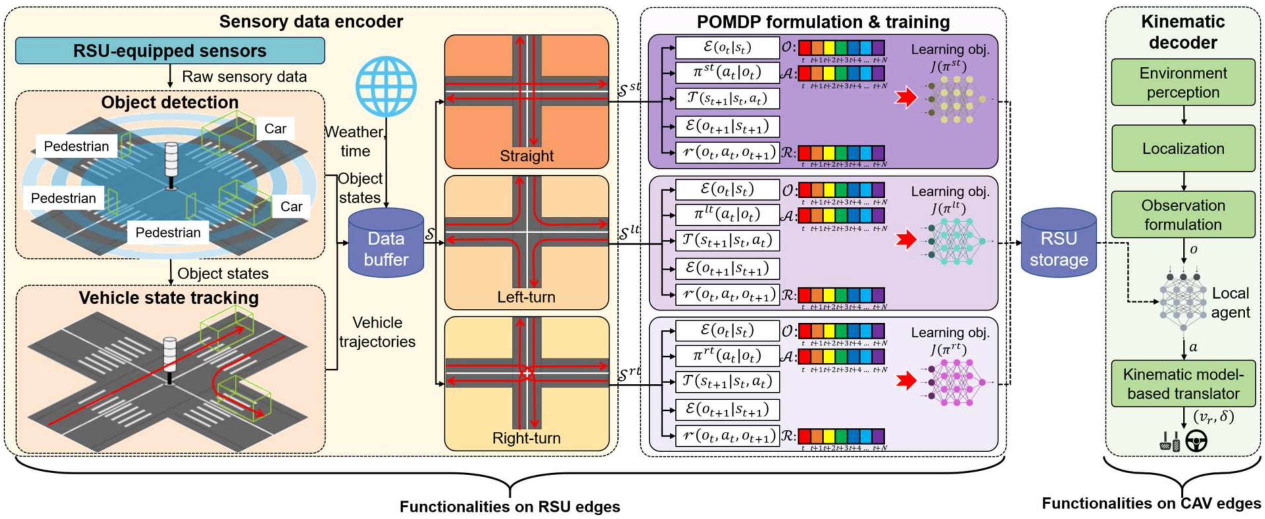


Figure 5.2: An overview of offline RL framework across RSU edge, CAV edge, and cloud plane.

on the RSU edge, and autonomous systems in CAVs only deploy the trained agent to achieve vehicle maneuvering. In particular, we introduce a sensory data encoder, a formulation of a partially observed Markov decision process (POMDP), and a kinematic-model-based vehicle maneuvering decoder to enhance the decision-making process.

As shown in Figure 5.2, the design of the offline RL framework also includes three key components: RSU edge, CAV edge, and the central cloud, aiming to realize the concept of LDT-assisted hybrid autonomous driving system depicted in Figure 5.1. This framework is developed with the following considerations:

- We use RSU-installed sensors and edge computing devices to formulate an object-level environmental perception, where the different non-stationary traffic participants, e.g., vehicles, pedestrians, and cyclists, can be identified and their immediate states are obtained. Thus, an LDT can be realized on the RSU edge.

- RSU edge is designed to extract passing-by vehicles' behaviors and learn how human drivers maneuver in the local area. Then generate various *local agents* under different conditions, including driving directions, time periods, and weather types.

- The maneuvering decoder, i.e., model inference, is placed on the vehicle edge to generate control signals, considering the safety-critical nature of vehicle maneuvering.

Therefore, our proposed offline RL framework effectively fulfills the conceptual design, unleashing the resources and capabilities of cloud and edge computing.

5.3.1 Sensory Data Encoder

The dynamic intersection environment is continuously monitored by RSU, which is capable of recognizing, detecting, and tracking all traffic participants within sensory range, covering the entire intersection area. We define $\mathcal{P}_t = \{p_1, p_2, \ldots, p_N\}$ as the set representing all detected traffic participants within the intersection at time step t. For each participant p_k in $\mathcal{P}_t$, its state is defined as follows:

$$\boldsymbol{s}_{t,k} = [c_k, l_k, w_k, x_k, y_k, \theta_k, v_k, v_{\text{lon},k}, v_{\text{lat},k}],$$

where c_k represents the category of the detected object, which could be a pedestrian, a vehicle, or a cyclist, while l_k and w_k represent its length and width. x_k and y_k capture the absolute position coordinates of the center point of p_k. The heading angle, i.e., yaw angle, is represented by θ_k. Variables v_k, $v_{\text{lon},k}$, and $v_{\text{lat},k}$ represent velocity, longitudinal velocity, and lateral velocity of p_k, respectively. The set $\mathcal{S}_t = \{\boldsymbol{s}_{t,1}, \boldsymbol{s}_{t,2}, \ldots, \boldsymbol{s}_{t,N}\}$ represents the state space at time step t describing the states of all traffic participants detected by RSU.

Based on real-time object detection, we track the vehicle trajectories to identify their direction of movement and categorize vehicles into three types: straight, left-turn, and right-turn. Thus, the state $\mathcal{S}$ can be divided into three subsets, i.e., $\mathcal{S}^{st}$, $\mathcal{S}^{lt}$, and $\mathcal{S}^{rt}$. This behavior classification ensures the directional diversity of *local agents* so that no matter which road an autonomous vehicle moves in and plans to move out, it can be provided with a feasible agent to guide it in an expected moving direction.

5.3.2 Formulation of POMDP

Since the amount of detected traffic participants N fluctuates over time, it is impractical to include all objects within the observation, particularly for those areas with heavy traffic. Thus, we use partially observed information to determine vehicles' actions. Such a problem can be formulated as a POMDP [176]. Offline RL in the context of a POMDP is formally described by a tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{E}, \mathcal{R}, \gamma)$ where $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{O}$ represent the spaces of states, available actions, and observations. $\mathcal{T}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ defines the probability of transitioning to state $\boldsymbol{s}_{t+1}$ from state $\boldsymbol{s}$ after taking action $\boldsymbol{a}_t$ that describes the dynamics of the system. $\mathcal{E}(\boldsymbol{o}_t|\boldsymbol{s}_t)$ is

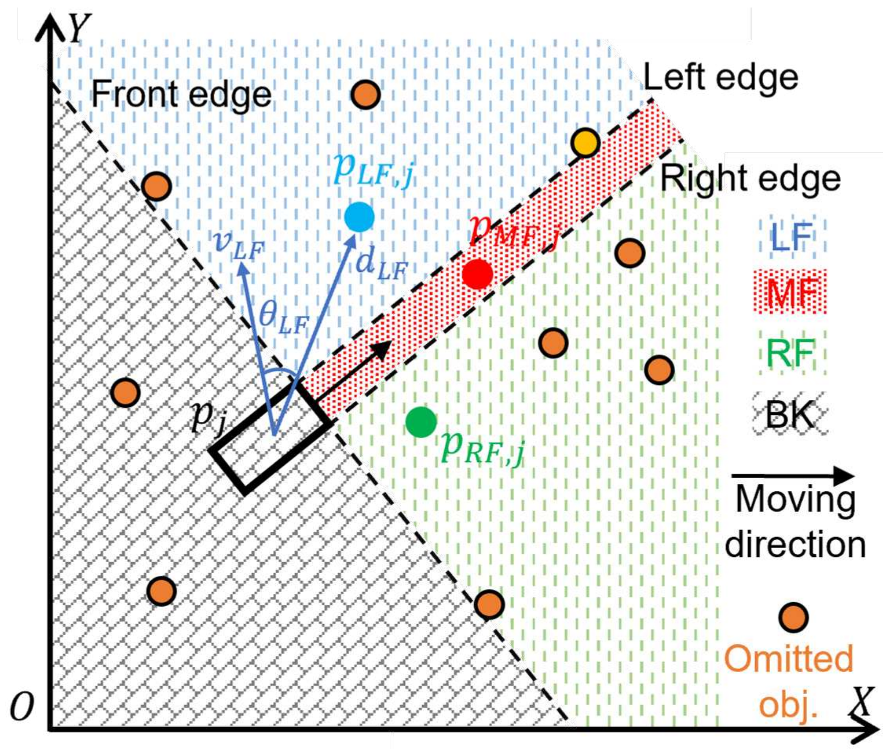


Figure 5.3: Illustration of observation space, as well as relative distances and velocity between observed vehicle with its surroundings.

an emission function defining the distribution from $\boldsymbol{s}_t$ to $\boldsymbol{o}_t$. $\mathcal{R}$ is the reward expressed with $r(\boldsymbol{o}_t, \boldsymbol{a}_t, \boldsymbol{o}_{t+1})$ and $\gamma \in (0, 1]$ is a scalar discount factor.

Observation: At a specific time step t, vehicle $p_j \in \mathcal{P}_t$ will be passing through the intersection, here, c_j is 'Vehicle'. The obstacles in the surrounding area, particularly along the vehicle's heading direction, are critical for road safety. We divide the surrounding area of the vehicle p_j into "left-front" (LF), "middle-front" (MF), "right-front" (RF), and "back"(BK) areas, based on the vehicle's frontal and lateral boundaries as depicted in Figure 5.3.

In the LF, MF, and RF areas, the closest objects to the vehicle p_j are denoted by $p_{\mathrm{LF},j}$, $p_{\mathrm{MF},j}$, and $p_{\mathrm{RF},j}$, respectively, as shown in Figure 5.3. At any given time t, the observation $\boldsymbol{o}_t \in O$ is given by:

$$\boldsymbol{o}_t = [\boldsymbol{s}_j, \boldsymbol{s}_{\mathrm{LF},j}, \boldsymbol{s}_{\mathrm{MF},j}, \boldsymbol{s}_{\mathrm{RF},j}],$$

where $\boldsymbol{s}_j$ is the state of vehicle p_j, and $\boldsymbol{s}_{\mathrm{LF},j}$, $\boldsymbol{s}_{\mathrm{MF},j}$, and $\boldsymbol{s}_{\mathrm{RF},j}$ represent the states of the nearest objects within the LF, MF, and RF areas, respectively. The design of the observation

$\boldsymbol{o}_t$ is based on the following considerations: *(i)* Human drivers make decisions based on their visual field, which excludes objects far away from ego vehicle or hidden in blind spots. Thus, from a causality perspective, we only consider visible and closest objects since they directly influence the decision-making of human drivers; *(ii)* Our system fuses onboard sensing and RSU perception data, ensuring that the CAV can detect and react to objects as soon as they enter the observation space, and *(iii)* given the variable amount of detected objects, considering only the nearest objects ensures a practical and computationally efficient observation space while still capturing the most immediate and impactful elements for safe driving.

Action: To effectively accommodate the diverse mechanical constraints of different vehicles, such as baseline length, steering limits, and acceleration capabilities, action $\boldsymbol{a}$ should be designed considering three requirements: *(i)* capabilities to reflect strategic driving decisions in the traffic environment, *(ii)* it can be derived from the vehicle's state s_j, and *(iii)* can be translated into the control space for different autonomous vehicles. Specifically, for the vehicle p_j at time t, the action $\boldsymbol{a}_t$ will be:

$$\boldsymbol{a}_t = [v_{x,j}, v_{y,j}, \dot{\psi}_j], \tag{5.1}$$

where v_x and v_y represent the vehicle's velocity components along the X-axis and Y-axis, respectively, and $\dot{\psi}$ represents the yaw rate. Here, the origin of the coordinate system is the location of the RSU. As such, the actions of all vehicles are represented in the same coordinate system. Thus the projection of velocity v_x and v_y onto this coordinate system becomes an insightful indicator of the vehicle's driving strategy.

Reward: To effectively evaluate the performance of driving strategies within our framework, the reward function is constructed as a linear combination of three critical components: road safety, traveling efficiency, and deviation from the road centerline.

$$r = \alpha_1 r_{\text{safety}} + \alpha_2 r_{\text{effi}} + \alpha_3 r_{\text{dev}}, \tag{5.2}$$

where α_1, α_2, and α_3 serve as coefficients to weight the safety reward, efficiency reward, and deviation reward, respectively. The initial values of these coefficients are determined based on domain knowledge and the characteristics of the target environment. For instance, in scenarios where safety is a top priority, α_1 is assigned a relatively larger value, while α_3 is set smaller in environments with wide, unstructured roads where strict adherence to the centerline is less critical. These weights are further refined through iterative training and evaluation. During this process, the convergence behavior of the training model and performance metrics are analyzed to identify and address suboptimal behaviors caused by inappropriate weight settings (e.g., excessively low values for α_1 leading to unsafe strategies). By systematically adjusting the weights, the reward function achieves a balance between safety and efficiency while ensuring stable and effective learning outcomes.

The safety component r_{safety} is defined as the potential time-to-collision (TTC) between the vehicle and the three closest objects. TTC is chosen as the primary safety metric because it is a comprehensive measure that captures multiple factors relevant to safety, including the relative position, relative velocity, and angle between the vehicle and surrounding objects. By focusing on TTC, the reward function directly addresses the critical aspect of collision risk, which is essential for ensuring safety in dynamic traffic scenarios. Its predictive nature enables proactive decision-making, as it quantifies how soon a collision may occur under the current conditions. For example, considering the closest object on the LF side of the vehicle $p_{\text{LF},j}$, we use $d_{\text{LF},j}$ and $v_{\text{LF},j}$ to represent the relative distance and relative velocity between the p_j and $p_{\text{LF},j}$, and $\theta_{\text{LF},j}$ to define the angle between $d_{\text{LF},j}$ and $v_{\text{LF},j}$, as shown in Figure 5.3. In a short time interval, it can be assumed that the vehicle and the object do not change their velocity, so the TTC can be expressed as:

$$\tau_{\text{LF},j} = \begin{cases} \frac{d_{\text{LF},j}}{v_{\text{LF},j}\cos\theta_{\text{LF},j}}, & \text{if } \theta_{\text{LF},j} \in [0, \frac{\pi}{2}), \\ +\infty, & \text{if } \theta_{\text{LF},j} \in [\frac{\pi}{2}, \pi]. \end{cases} \tag{5.3}$$

When the value of $\theta_{\text{LF},j}$ is smaller than $\pi/2$, we calculate the potential TTC based on relative distance velocity. If $\theta_{\text{LF},j}$ is larger than $\pi/2$, indicating that p_j and $p_{\text{LF},j}$ are moving away from each other, then the TTC is set to infinity. Similar to the calculation of $\tau_{\text{LF},j}$ in (5.3), we can obtain $\tau_{\text{MF},j}$ and $\tau_{\text{RF},j}$ for the closest objects on the MF and RF sides. r_{safety} is then determined by:

$$r_{\text{safety}} = \sum_{i=\text{LF,MF,RF}} \beta_i \frac{\min[\tau_{i,j}, \tau_{\text{thre}}]}{\tau_{\text{thre}}}, \tag{5.4}$$

where τ_{thre} is a predefined threshold that serves as a boundary to evaluate the urgency of potential collisions. The value of τ_{thre} is set to 3 seconds based on the Japanese traffic rule [178]. β_i is a parameter that balances the importance of the three nearest objects. For example, for a vehicle making a left turn, its expected trajectory is in the LF region, thus β_{LF} is assigned a larger value, while β_{RF} is with a smaller value, i.e., $\beta_{\text{LF}} > \beta_{\text{MF}} > \beta_{\text{RF}}$. As for a straight-moving vehicle, β_{MF} is assigned a larger value, i.e., $\beta_{\text{MF}} > \beta_{\text{LF}} = \beta_{\text{RF}}$.

The efficiency reward component r_{effi} is defined as the optimal travel speeds normalized by the maximum speed,

$$r_{\text{effi}} = \begin{cases} \frac{v_j}{v_{\max}}, & \text{if } v_j \leq v_{\max}, \\ -\frac{v_j - v_{\max}}{v_{\max}}, & \text{if } v_j > v_{\max}, \end{cases}$$

where $v_{\max}$ represents the speed limit. When $0 \leq v_j \leq v_{\max}$, r_{effi} is positive to encourage velocity maintenance near the speed limit. When $v_j > v_{\max}$, r_{effi} becomes negative for the penalty of overspeeding.

Lastly, the reward for minimizing deviation from the lane's centerline r_{dev}, is designed to incentivize precise lane-keeping by penalizing deviations from the centerline:

$$r_{\text{dev}} = 1 - \frac{2\Delta}{w},$$

where Δ and w represent the lateral displacement from the centerline and the average lane width, where w can be obtained based on static high-definition (HD) maps.

As for the training processes, we introduce the twin delayed deep deterministic policy gradient with behavior cloning (TD3+BC) algorithm [179] to obtain offline policy π, since *(i)* it effectively balances exploration and exploitation by combining RL with behavior cloning (BC), ensuring robust policy learning from pre-collected driving data, *(ii)* it reduces overestimation bias in Q-value estimates, which is crucial for accurately evaluating actions in the offline setting of local driving data, and *(iii)* it is computationally efficient and easy to implement, which is essential for training *local driving agents* with limited computational resource of RSU edge. Basically, TD3+BC's policy π is updated with the deterministic policy gradient with a BC term:

$$\pi = \arg\max_{\pi} \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim D} \left[\lambda Q(\boldsymbol{s}, \pi(\boldsymbol{s})) - (\pi(\boldsymbol{s}) - \boldsymbol{a})^2\right], \tag{5.5}$$

where $Q(\boldsymbol{s}, \pi(\boldsymbol{s}))$ represents the Q-value function, and λ is a hyperparameter that controls the weight of the Q-value estimation against the behavior cloning (BC) term.

5.3.3 Kinematic Decoder for Vehicle Maneuvering

In order to use *local agents* on CAV edge, it is necessary to formulate observation space based on environment perception (generating $\boldsymbol{s}_{LF,j}$, $\boldsymbol{s}_{MF,j}$, and $\boldsymbol{s}_{RF,j}$) and localization (reflecting $\boldsymbol{s}_j$) at first. Then we can use trained agents to output actions. However, action space in (5.1)

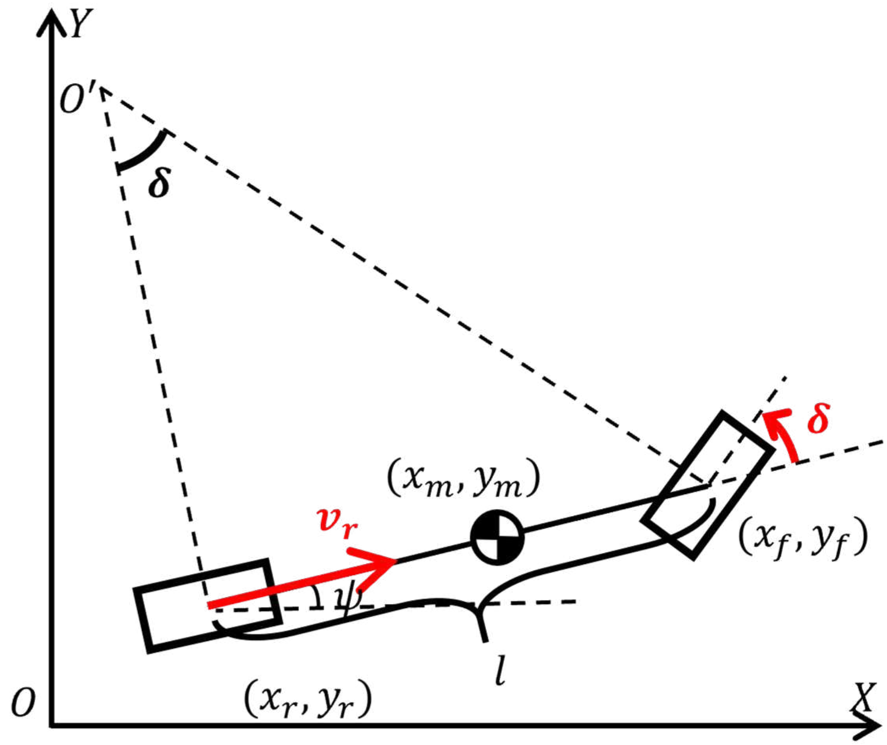


Figure 5.4: Illustration of kinematic bicycle model.

cannot be directly used for vehicle control, hence we introduce a kinematic bicycle model to simplify car-like vehicle dynamics, as shown in Figure 5.4, and to translate the action space into control space. The ego vehicle is modeled with front-wheel steering and rear-wheel drive as an example of the kinematic decoder design. For vehicles with different configurations, such as front-wheel drive, the corresponding kinematic model (e.g., see [180]) should be applied to achieve vehicle control. We use $(x_{\mathrm{f}}, y_{\mathrm{f}})$, $(x_{\mathrm{r}}, y_{\mathrm{r}})$ and $(x_{\mathrm{m}}, y_{\mathrm{m}})$ to represent the coordinates of centers of the front axle, rear axle, and the center point of the baseline. v_{f} and v_{r} are the velocities of front and rear wheels. The yaw angle and steering angle are represented by ψ and δ, respectively. l_{base} is the length of the vehicle baseline. At the center of the rear axle, the velocity v_{r} can be expressed with its projections on the X-axis and Y-axis:

$$v_{\mathrm{r}} = \dot{x}_{\mathrm{r}} \cos\psi + \dot{y}_{\mathrm{r}} \sin\psi. \tag{5.6}$$

We use the position of the middle point to express the position of the rear axle:

$$\begin{cases} x_{\mathrm{r}} = x_{\mathrm{m}} - \dfrac{l_{\mathrm{base}}}{2} \cos \psi, \\ y_{\mathrm{r}} = y_{\mathrm{m}} - \dfrac{l_{\mathrm{base}}}{2} \sin \psi. \end{cases} \tag{5.7}$$

The differential form of (5.7) is:

$$\begin{cases} \dot{x}_{\mathrm{r}} = \dot{x}_{\mathrm{m}} + \dfrac{l_{\mathrm{base}}\dot{\psi}}{2} \sin \psi, \\ \dot{y}_{\mathrm{r}} = \dot{y}_{\mathrm{m}} - \dfrac{l_{\mathrm{base}}\dot{\psi}}{2} \cos \psi. \end{cases} \tag{5.8}$$

Here, we note that $\dot{x}_{\mathrm{m}}$ and $\dot{y}_{\mathrm{m}}$ correspond to the v_x and v_y in the action space. Based on (5.6) and (5.8), the rear tire velocity can be expressed with the v_x and v_y:

$$v_{\mathrm{r}} = v_x \cos \psi + v_y \sin \psi. \tag{5.9}$$

At the center of the front and rear axles, the kinematic constraints are:

$$\begin{cases} \dot{x}_{\mathrm{f}} \sin (\psi + \delta) - \dot{y}_{\mathrm{f}} \cos (\psi + \delta) = 0, \\ \dot{x}_{\mathrm{r}} \sin \psi - \dot{y}_{\mathrm{r}} \cos \psi = 0. \end{cases} \tag{5.10}$$

Based on (5.6) and (5.10), we can obtain:

$$\begin{cases} \dot{x}_{\mathrm{r}} = v_{\mathrm{r}} \cos \psi, \\ \dot{y}_{\mathrm{r}} = v_{\mathrm{r}} \sin \psi. \end{cases} \tag{5.11}$$

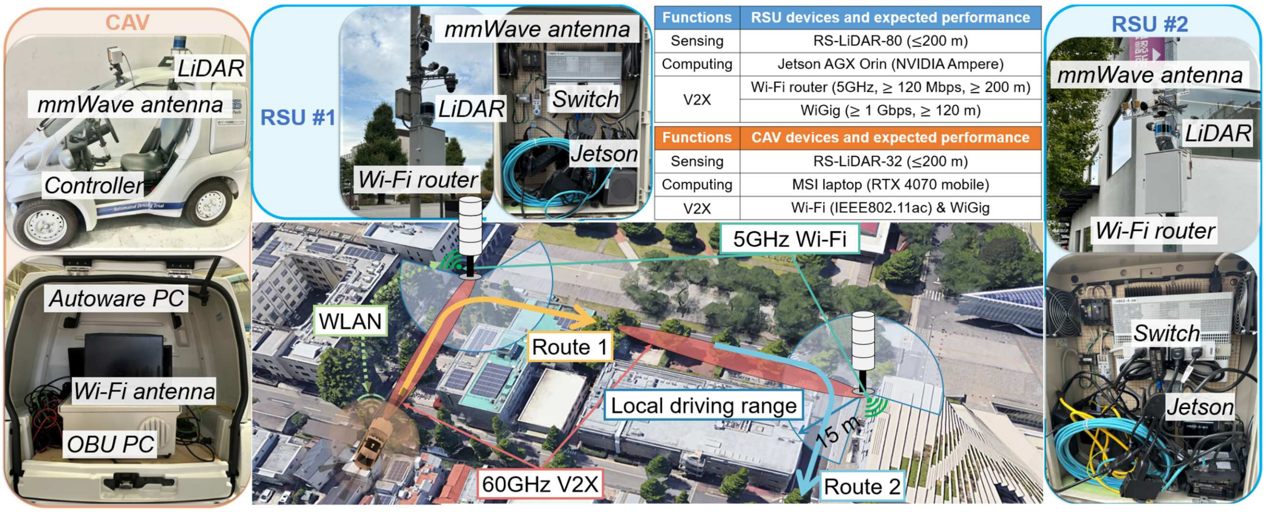
According to the relative positions of the front and rear axles, the position of the front axle is:

$$\begin{cases} x_{\mathrm{f}} = x_{\mathrm{r}} + l_{\mathrm{base}} \cos \psi, \\ y_{\mathrm{f}} = y_{\mathrm{r}} + l_{\mathrm{base}} \sin \psi. \end{cases} \tag{5.12}$$

Based on (5.10)-(5.12), the steering angle can be expressed with vehicle yaw rate:

$$\delta = \arctan \frac{l_{\mathrm{base}}\dot{\psi}}{v_{\mathrm{r}}}. \tag{5.13}$$

For the autonomous driving motion control, (v_{r}, δ) is regarded as the control space. According to (5.9) and (5.13), we translate the action space to the control space, thereby facilitating motion control for autonomous vehicles.



Functions	RSU devices and expected performance
Sensing	RS-LiDAR-80 (≤200 m)
Computing	Jetson AGX Orin (NVIDIA Ampere)
V2X	Wi-Fi router (5GHz, ≥ 120 Mbps, ≥ 200 m)
	WiGig (≥ 1 Gbps, ≥ 120 m)
Functions	**CAV devices and expected performance**
Sensing	RS-LiDAR-32 (≤200 m)
Computing	MSI laptop (RTX 4070 mobile)
V2X	Wi-Fi (IEEE802.11ac) & WiGig

Figure 5.5: An overview of the hybrid autonomous driving testing field with specifications of hardware deployment.

5.4 Real-world Prototype and Experiment Settings

In this section, we explain how to implement our designed system in a real-world prototype, from the perspective of both hardware and software deployments. Then, we introduce some important settings in HiL and PoC experiments.

5.4.1 Hardware Deployment and Software Installation

According to the system design and the requirements in Section II, our hybrid autonomous driving testing field is designed as illustrated in Figure 5.5. Within this setting, we install two RSUs at two different intersections. Each RSU is equipped with an 80-layer LiDAR and an NVIDIA Jetson. The LiDAR captures raw data, i.e., dynamic point clouds, from the physical environment. The Jetson is tasked with executing various functional modules, such as object detection and tracking, thereby creating object-level LDT at the RSU edge. As shown in Figure 5.6, we demonstrate an example of intersection congestion in the testing field and its corresponding LDT. A CAV is mounted with a 32-layer LiDAR to monitor its surroundings. An Autoware PC processes the LiDAR data, enabling environmental perception, localization, motion planning, and motion control. The control signals are then transmitted to the onboard unit (OBU) to facilitate autonomous driving. The communications system in the hybrid autonomous driving platform is built on a heterogeneous V2X network. For perception

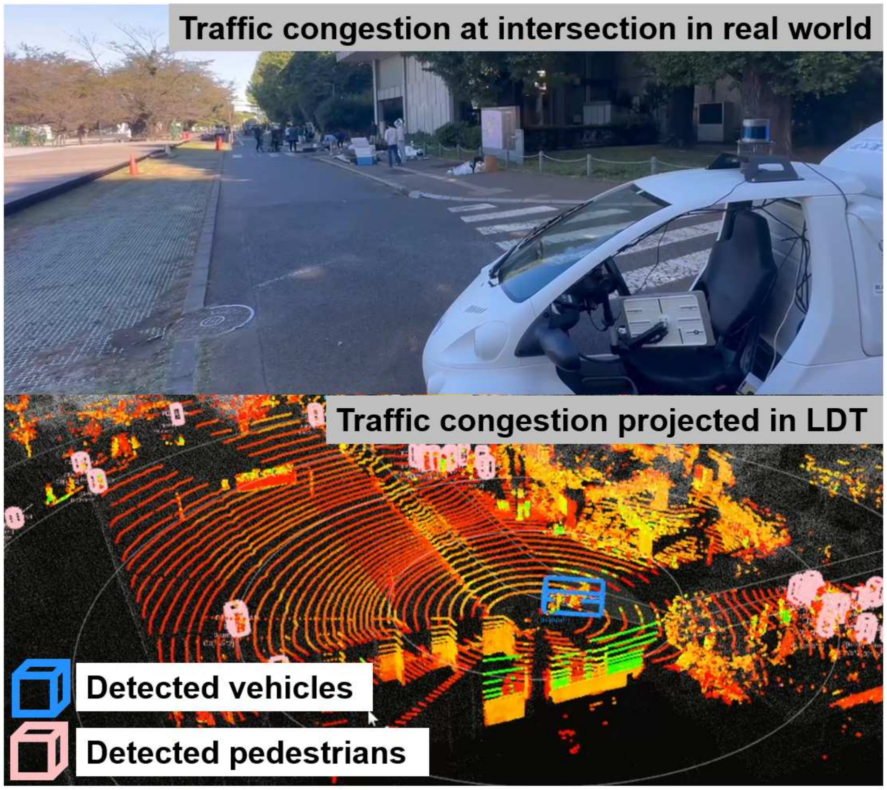


Figure 5.6: An illustration of congestion at an intersection in the real world and LDT.

information sharing, we employ Wi-Fi modules working at 5 GHz to let CAV receive cooperative perception data from RSUs. As for *local driving agents* transfer, given the substantial data size of AI models, we use 60 GHz millimeter wave (mmWave) communications to enable efficient and high-speed data transmission.

Our platform's software system is primarily based on Autoware.Universe [155] and version 2 of robot operating system (ROS2). Autoware supports the detection and tracking of traffic participants within the RSU sensor range. The detection module employs the CenterPoint framework [93], which can detect, identify, and visualize 3D objects from LiDAR point clouds in real time. Subsequently, a multiple object tracking (MOT) module [97] assigns IDs to the detected objects so that we can record each object's historical trajectory and classify them based on moving directions. As for the autonomous driving system, we also apply a normal distributions transform (NDT) algorithm, OpenPlanner, and a pure pursuit algorithm for localization, motion planning, and motion control modules, respectively. Detailed information about the hardware and software systems in the test field can be found in Chapter 3.

5.4.2 Simulation and Field Trial Setups

Following the discussion in Section III, we first utilize two RSUs to monitor their surrounding traffic flows and generate right-turn *local driving agents*, named RSU #1 local agent and RSU #2 local agent. Then, we conduct PoC tests and HiL simulations to validate the proposed system. As depicted in Figure 5.6, we define two right-turn routes (Route #1 and Route #2) that respectively pass by RSU #1 and RSU #2-installed intersections. We also define a 'local driving range' (15 m). When CAV enters the 'local driving range', CAV will adopt *local driving agents* as a replacement for its inherent motion planning and motion control modules. The local driving range of 15 m is determined based on the road topology of the test field and vehicle speed limitation of 20 km/h, where the 15 m range is sufficient to cover the entire

Table 5.2: PoC and HiL setups.

Specifications	**Value/Description**
Local driving range	15 m (CAV adopts local agents)
Autonomous vehicle actions	CAV adopts local driving agents when entering the defined local driving range
Traffic density conditions	Low (<3 objects), Middle (3-6 objects), and High (>6 objects)
Agent types	RSU #1 Local Agent, RSU #2 Local Agent, and Inherent Autoware Agent
PoC experiments	
Field trial count	9 tests (3 tests per scenario, conducted on Route #1 only, using 3 types of agents)
Used RSU(s)	RSU #1
Driving route(s)	Route #1
Communications distance	30 m (V2I communications range)
HiL simulations	
Simulation count	180 tests (10 tests per scenario, 3 traffic conditions for both routes, using 3 types of agents)
Used RSU(s)	RSU #1 and RSU #2
Driving route(s)	Route #1 and Route #2

intersection. For larger intersections or scenarios involving higher vehicle speeds, the local driving range should be redefined accordingly. In addition, to test the autonomous driving performance in different traffic conditions, we categorize the traffic into 3 types: low density (the number of objects within the intersection area is less than 3), middle density (ranging from 3 to 6), and high density (larger than 6). Some detailed settings in the simulation and field trial are described as follows and summarized in Table 5.2.

5.4.2.1 HiL simulation

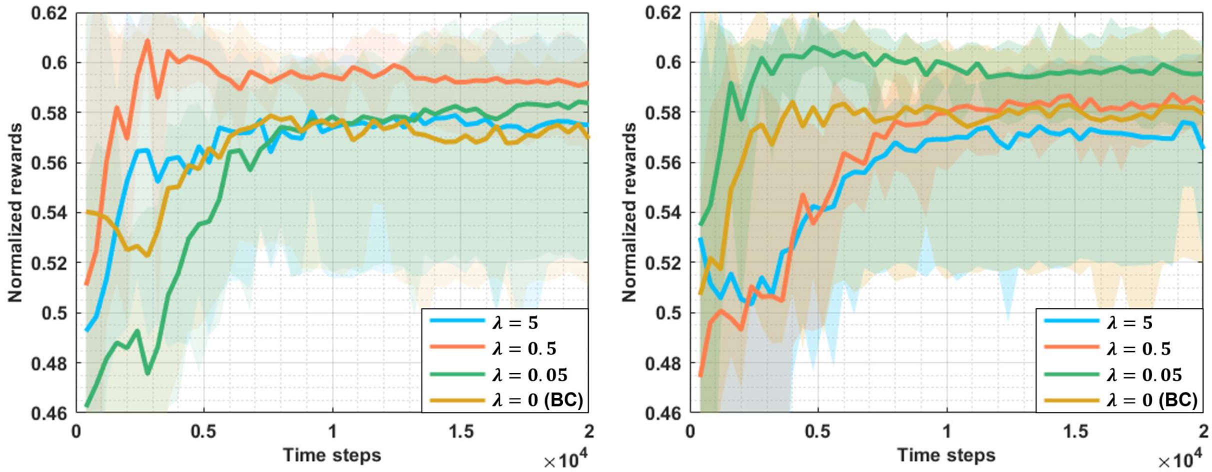
In order to repeatedly evaluate the performance of our system in autonomous driving, we conducted HiL simulations using real traffic data perceived by RSUs and a simulated vehicle. We utilized 3 types of agents (RSU #1 local agent, RSU #2 local agent, and the inherent Autoware agent) to drive the simulated CAV through Route #1 and Route #2 under 3 different traffic conditions (low, middle, and high density). Each scenario was tested 10 times, resulting in a total of 180 experiments.

5.4.2.2 PoC field trial

Our experimental environment involves public traffic and the intersection at Route #2 presents highly complex traffic conditions. Therefore, for safety reasons, we conduct field tests only on Route #1. Here, we define a 'communications distance' (30 m), within which the V2I communications is activated that RSU will disseminate *local driving agents* through mmWave communications and provide processed data cooperative perception services through the WLAN network. We conducted 9 experiments in total where we employed 3 types of agents (RSU #1 local agent, RSU #2 local agent, and the inherent Autoware agent) to navigate the CAV through Route #1 under 3 different traffic conditions (low, middle, and high density).

5.5 Experiment Results and System Evaluation

In this section, we first discussed the offline RL training results. Then, to evaluate the crucial KPIs of the implemented hybrid autonomous driving system, we empirically measured several communications metrics in the prototype. Finally, we showed both the HiL and PoC test results in terms of safety, efficiency, and real-time performance to validate the improvement brought to autonomous driving.



(a) RSU #1-installed intersection traffic.

(b) RSU #2-installed intersection traffic.

Figure 5.7: Offline RL training results on offline traffic data: Normalized rewards by varying λ value over 10 random seeds.

5.5.1 Performance Comparison of Offline RL Algorithms

We provide the training performance in terms of normalized rewards in Figure 5.7, where the solid lines and dimmed areas represent the mean and confidence interval over 10 random seeds, respectively. The maximum time step to run our customized environment is set to 20,000, with evaluations conducted every 400 time steps. Remarkably, the TD3+BC algorithm can always achieve convergence with varying hyperparameter λ in (5.5), which demonstrates the effectiveness of our dataset and the feasibility of our POMDP formulation.

For RSU #1-installed intersection, the convergence result with $\lambda = 0.05$ is around 0.59, surpassing that with other λ values. As for RSU #2, the convergence performance is optimal when $\lambda = 0.5$. Therefore, in subsequent simulation and field trial tests, models of TD3+BC algorithms trained with $\lambda = 0.05$ and $\lambda = 0.5$ are regarded as RSU #1 and RSU #2 local agents, respectively.

5.5.2 System Evaluation Considering Communications Issues

Some communications metrics are measured in PoC tests, as shown in Table 5.3. Compared to the required KPIs listed in Table I, Wi-Fi-based cooperative perception and mmWave communications-enabled agent downloading services can both meet the specified requirements. As for the agent downloading service, we first measure the data size of agents trained with TD3+BC algorithms, which ranges between 11 and 13 MB, fitting within the defined range of 2 to 40 MB. The data rate achieved during the downloading process is at least 0.711 Gbps, averaging 1.24 Gbps, greatly surpassing the peak data rate requirement of 100 Mbps. The E2E latency for downloading agents is observed to be no more than 146 ms with an average latency of 83.9 ms, which is also well within the acceptable range of 500 to 1000 ms as defined in the 3GPP TR 22.874. The PoC tests are also conducted on NVIDIA Jetson system equipped with an ARM Cortex-A78AE CPU (12 cores), 32 GB of memory, and Ubuntu 20.04 operating system, over a duration of 10 minutes. Here, we note that mmWave communications may face challenges in extreme environments, such as rain or obstructions, which can affect communications stability. The heterogeneous V2X communications utilized in our system can mitigate such challenges by switching to Wi-Fi-based V2I or cellular networks to ensure reliable communications.

As for cooperative perception, since ROS2 facilitates seamless communications among distributed computers within a local area network (LAN), the CAV can easily access all ROS2-defined messages from the RSU edge using user datagram protocol (UDP). In this study, the CAV is designed to subscribe to object detection messages generated in nearby RSUs to facilitate cooperative perception services. We evaluate the communications reliability using the observed packet delivery ratio (PDR), which achieves a minimum of 96.1%, exceeding the 90% reliability threshold specified for the SSMS use case. Furthermore, the E2E latency and data rate are observed to be less than 8.51 ms and higher than 25.9 Mbps, also well-satisfying

Table 5.3: Communications performance of agent downloading and cooperative perception services.

Services	**Agent downloading (mmWave)**	**cooperative perception (Wi-Fi)**
Reliability (%)	-	$\geq$ 96.1 (97.3 on average)
Latency (ms)	$\leq$ 146 (83.9 on average)	$\leq$ 8.51 (6.57 on average)
Data rate (Mbps)	$\geq$ 711 (1240 on average)	$\geq$ 25.9 (32.1 on average)
Data size (MB)	11$\sim$13	-

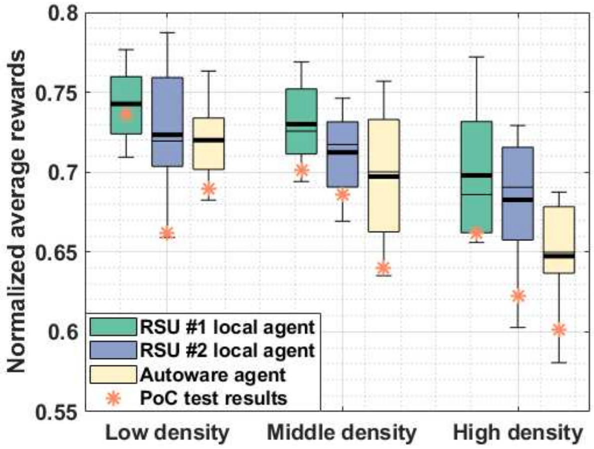
the 10 ms latency and 25 Mbps peak data rate requirements. These results confirm that the proposed system supports cooperative perception and agent downloading services to fulfill the stringent KPIs for SSMS and AI/ML based automotive networked systems use cases.

5.5.3 HiL and PoC Test Results

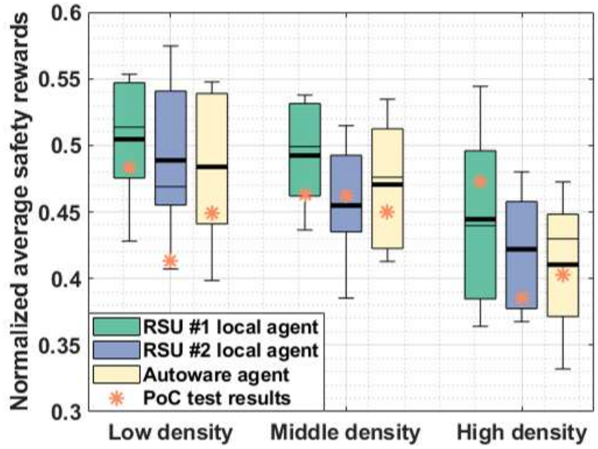
As discussed in Section IV.B, we conduct both HiL and PoC experiments in RSU #1-installed intersection, whose results are shown in Figure 5.8, where box plots are employed to visually represent the distribution of combined rewards (defined in (5.2)), safety rewards (defined in (5.4)), and traveling time attained by various agents during autonomous driving process, and the mean value is delineated using a bold solid line. We also mark the performance of PoC field tests in Figure 5.8 as orange star marks. As for RSU #2-installed intersection, we only showcase the HiL simulation results in Figure 5.9.

According to the data presented in Figure 5.8(a), it is evident that RSU #1 local agent shows better overall performance than RSU #2 local agent and Autoware agent in Intersection #1, no matter within simulation or field trial. As illustrated in Figure 5.9(a), it also can be found that RSU #2 local agent always drives the vehicle better than RSU #1 local agent and Autoware agent in Intersection #2. Such outcomes validate the effectiveness of our system design and offline RL formulation from the following two perspectives: *(i)* The overall superior performance of the designed local agents, in comparison to the general algorithm, suggests that the proposed system has effectively enhanced autonomous driving at traffic intersections, and *(ii)* The different performance of the two trained agents at two intersections demonstrates the uniqueness of each trained agent, suggesting that models trained on local traffic flow data perform well only when applied to their respective local environments. This indicates that our system effectively captures and learns intersection-specific local strategies from human driving behaviors.

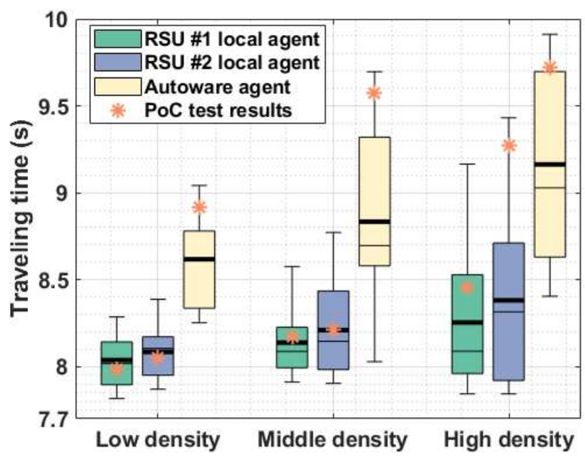
To quantify the evaluation of safety, we employ the safety reward function expressed in (5.4). Although the Autoware agent adopts a relatively conservative and safety-oriented driving strategy, which refrains from actively overtaking obstacles and instead opts to wait for their leaving, the average safety rewards depicted in Figure 5.8(b) and Figure 5.9(b) show that the trained agents exhibit safety improvements. For Intersection #1, compared to the conventional autonomous driving algorithm, i.e., Autoware agent, the local agent exhibits a safety promotion of 5.8% (on average) and 9.3% within HiL simulation and PoC experiments, respectively. As for Intersection #2, its local agent also demonstrates better safety measures



(a) Distribution of normalized combined rewards.



(b) Distribution of normalized safety rewards.



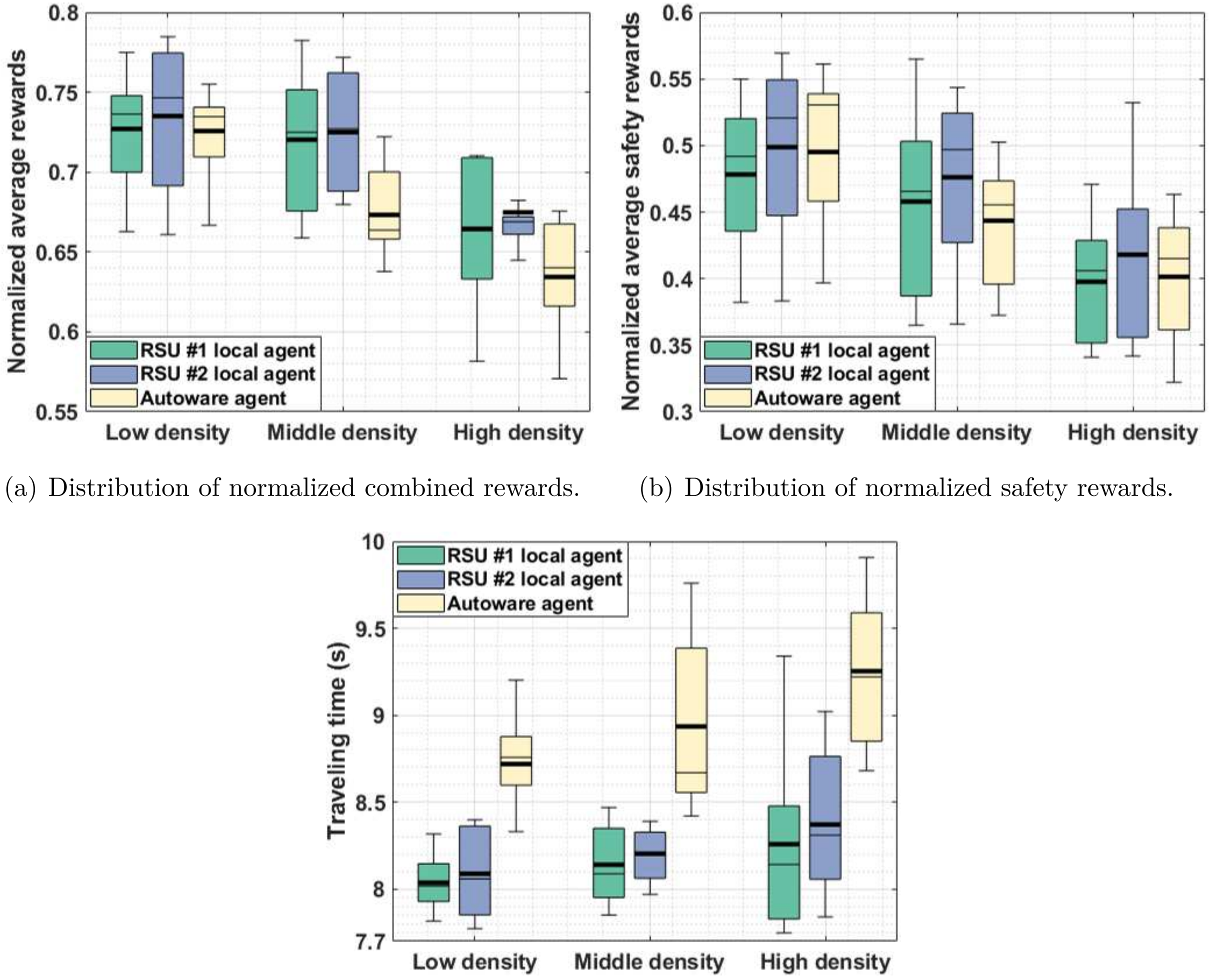
(c) Distribution of traveling time

Figure 5.8: HiL simulation and PoC test results of applying RSU #1 local agent, RSU #2 local agent, and Autoware agent in RSU #1-installed 'local driving range'.

with a 4.1% improvement on average.

For evaluating driving efficiency, we show the distribution of traveling time within defined 'local driving range' to offer an intuitive representation of efficiency performance. Figure 5.8(c) and Figure 5.9(c) illustrate that the local agents exhibit considerable enhancements in driving time reduction. RSU #1 local agent can reduce travel time by 8.9% on average according to simulation results, and up to 14.6% in field tests. In the HiL experiments at Intersection #2, it can be found that RSU #2 local agent also decreases the total moving time by 9.1%.

Finally, we evaluate the real-time performance of proposed local driving agents and some



(a) Distribution of normalized combined rewards.

(b) Distribution of normalized safety rewards.

(c) Distribution of traveling time

Figure 5.9: HiL simulation results of applying RSU #1 local agent, RSU #2 local agent, and Autoware agent in RSU #2-installed 'local driving range'.

update frequencies of some crucial functional blocks. The CAN controller of our CAV is designed to operate at 10 Hz, requiring the autonomous driving software system to send control signals every 100 ms. The low-level motion control module of inherent Autoware system also operates at 10 Hz, ensuring that the Autoware can control our autonomous vehicle in real-time effectively. For our hybrid autonomous driving system, we replace the motion planning and motion control functions at intersections with an online execution (model inference) of local driving agents. The input information for the model inference includes the vehicle's onboard perception data, processed cooperative perception information shared by RSUs, and vehicle localization. All input data updates at a frequency larger than 30 Hz. In addition, the average computation time for model inference is measured at 38.4 ms. Therefore, the output frequency of the local agent can be set to 10 Hz to ensure real-time input and output, thus enabling vehicle control in real time. Consequently, our designed hybrid autonomous driving system meets the real-time requirements for autonomous driving.

5.6 Summary

In this Chapter, we proposed a novel LDT-assisted hybrid autonomous driving system, specifically designed for traffic intersections to improve road safety and driving efficiency. We addressed the challenge of the intricate and dynamic nature of intersections, where road topology, traffic volume, and complex traffic conditions necessitate a local decision-making approach. By leveraging RSUs equipped with sensory and edge computing capabilities, the system continuously monitors traffic, extracts human driving knowledge, and generates intersection-specific local driving agents through an offline RL framework. This method enables vehicles to adopt local driving agents generated from accumulated driving behavior data, particularly when passing through RSU-installed intersections.

When CAVs pass through RSU-installed intersections, RSUs can provide them not only with real-time cooperative perception services to broaden onboard sensory horizons but also with local driving agents to support safe and efficient driving in local areas. Extensive PoC field trials and HiL simulations validate the effectiveness of the system. The results demonstrate that our hybrid system outperforms the conventional Autoware algorithms, achieving average safety improvements of up to about 10% and travel time reductions of up to about 15%.

Additionally, the system meets stringent real-time performance requirements, with communications metrics showing peak latencies of 8.51 ms for cooperative perception services and 146 ms for local agent downloading, aligning with 3GPP requirements. These results confirm that integrating LDT-based *local driving strategies* can significantly enhance autonomous driving at complex urban intersections.

Chapter 6

Final Remarks and Future Work

6.1 Summary of the thesis

- Chapter 1: It reviews the history and current states of autonomous driving and digital twin (DT) technologies, highlighting the benefits of their convergence and integration. After revealing the research target "smart mobility digital twin (SMDT)", it underscored the limitations of existing mobility DTs and current vehicle-road collaboration methods. At the end, it summarized the author's efforts and contributions towards the research target.

- Chapter 2: It first provided a comprehensive overview of modular autonomous driving systems (ADS), including navigation, sensing, environmental perception, localization, prediction, local planning, motion control, and vehicle actuators. In the second part, it introduced the main techniques in the establishment of mobility DTs. Finally, it pointed out the challenges ahead of the successful transition from mobility DT to SMDT.

- Chapter 3: It introduced the SMDT platform, bridging the gap between physical and cyber worlds by utilizing cloud and edge computing resources, enabling real-time traffic monitoring, decision-making, and system optimization. It first provided the motivation for introducing SMDT to the autonomous driving field. Then, it presented the high-level conceptual architecture of the SMDT platform, detailing its integration of roadside unit (RSU) edge, connected and automated vehicle (CAV) edge, and a central cloud. Finally, it described the prototype system and its deployment in a real-world environment, highlighting the hardware and software components used to create real-time traffic DTs.

- Chapter 4: In order to effectively presented the design and implementation of a CAV navigation system based on the SMDT platform. The CAV navigation system aims to enhance traffic management by integrating real-time dynamic traffic data, V2X communication, and advanced route planning mechanisms. First, it outlineed the motivation for developing a navigation system tailored to address the limitations of existing autonomous vehicle frameworks. Then, it introduced the high-level design of the CAV navigation system, emphasizing its event-triggered route planning strategy and real-time adaptability. It also demonstrated a proof-of-concept (PoC) implementation within the Sci Tokyo Smart Mobility Field, showcasing the system's ability to respond to real-world traffic events. Finally, it evaluated the system's performance through large-scale simulations, focusing on traffic efficiency, safety improvements, and communication metrics.

- Chapter 5: It proposesed a novel local digital twin (LDT)-assisted hybrid autonomous driving system, specifically designed for traffic intersections to improve road safety and driving efficiency. Firstly, it outlined the motivations for developing LDTs, emphasizing their role in augmenting vehicle-road collaboration for safer and more efficient intersection management. Secondly, it presented the architecture of the LDT-assisted hybrid autonomous driving system, detailing the roles of RSUs, CAVs, and the central cloud in achieving localized decision-making and real-time control. It also described the offline reinforcement learning (RL) framework used to extract human driving knowledge and generate intersection-specific driving agents, highlighting its integration with the proposed system. Then, it explained the hardware deployment and software implementation of the system in a real-world testing environment, providing technical details about RSUs, CAVs, and communication infrastructures. Finally, it presented the results of PoC experiments and hardware-in-the-loop (HiL) simulations, showcasing the system's improvements in safety, efficiency, and real-time responsiveness at intersections.

Moreover, the unique contributions of Chapter 3, Chapter 4, and Chapter 5 works, compared to the existing mobility DTs and vehicle-road collaboration methods, can be showcased through Table 6.1 and Table 6.2.

Table 6.1: Comparison of related works of Mobility DT

Ref.	Key components			SAE level	DT functions		Validation methods	Objectives (use cases)
	Cloud	Vehicles	RSUs		Modeling	Service		
[35]	√	√	√	L2	√	√	Field test/ Simulation	Personalized adaptive cruise control (P-ACC).
[42]	√	√	√	L2	√		Field trial	Cooperative perception to create traffic DT.
[43]	√		√		√		Field trial	Cooperative vehicle-infrastructure system.
[46]		√		L4	√		Field trial	Validation framework for autonomous driving.
[50]	√	√				√	Simulation	Path planning considering unexpected events.
[51]	√	√	√		√	√	Simulation	Traffic light control system.
[54]	√	√		L2	√	√	Simulation	Cooperative driving at intersections.
[55]	√	√			√		Sandbox	Validation of multi-vehicle cooperation.
Chapter 3 and 4	√	√	√	L4/L5	√	√	Field test/ Simulation	SMDT-based CAV navigation system.

Table 6.2: Comparison with related works of vehicle-road collaboration

Ref.	Key components: Cloud	Key components: Vehicles	Key components: RSUs	SAE level	Unique traffic features	Validation methods	Objectives (use cases)
Cooperative perception-related studies							
[57]		√	√	-		Simulation	Generation of future BEV occupancy map.
[58]		√	√	-	√	Dataset	Early prediction of intersection movement and warning.
[59]		√	√	-	√	Dataset	Adaptive construction of a dynamic perception representation.
[67]		√	√	-		Simulation	Intersection traffic safety framework for collision avoidance.
Cooperative driving-related studies							
[68]		√	√	-		Simulation	Federated learning framework to reduce idling time.
[69]			√	-		Simulation	Traffic signal control system to reduce idling time.
[70]		√	√	L4/L5		Indoor robots	Intersection coordination framework for trajectory planning.
DT-based studies							
[71]		√	√	-		Field test	Highway DT for CAV perception enhancement.
[72]	√	√		L2		Field test	DT-based ramp merging framework.
[54]	√	√		L2		Simulation	DT-assisted FIFO framework at intersections
Chapter 5	√	√	√	L4/L5	√	Field test/simulation	Hybrid autonomous driving with local driving strategies

6.2 Suggested future works

In the future, large-scale simulations, deployments, and evaluations are needed to fill in the remaining gaps between the proposals and experiments. Moreover, the proposed systems and technologies have the potential to make significant contributions to several promising fields. We plan to make some explorations in the following fields.

6.2.1 City-wide SMDT Deployment

The implementation of the SMDT at a city-wide scale presents a significant opportunity to revolutionize urban transportation systems. While the current focus has been on developing the foundational components of the SMDT, deploying it across an entire city involves addressing challenges related to scalability, interoperability, and data management.

Future work in this area should concentrate on:

1. *Scalability solutions:* Developing algorithms and system architectures that can handle the vast amount of data generated by numerous vehicles, infrastructure sensors, and users within a city. This includes optimizing cloud and edge computing resources to ensure real-time data processing and response.

2. *Interoperability standards:* Establishing common standards and protocols to enable seamless communication and data exchange between different stakeholders, including various vehicle manufacturers, infrastructure providers, and service operators. This ensures that all components of the SMDT can work together cohesively.

3. *Data privacy and security:* Implementing robust security measures to protect sensitive data and ensure user privacy. As the SMDT collects and processes extensive amounts of information, safeguarding this data against cyber threats is paramount.

By focusing on city-wide deployment, future research can contribute to creating smarter, more efficient urban transportation networks that benefit all city inhabitants.

6.2.2 Multi-Vehicle Cooperative Driving System

The advancement of autonomous vehicles offers the potential for cooperative driving systems where multiple vehicles communicate and collaborate to optimize traffic flow and enhance safety. Integrating such systems within the SMDT can significantly improve the performance of autonomous vehicles and the overall transportation network.

Areas for future exploration include:

1. *Cooperative maneuvering algorithms:* Developing algorithms that enable vehicles to coordinate actions such as lane changes, merging, and intersection crossing. These algorithms should account for real-time data from other vehicles and infrastructure to make safe and efficient decisions.

2. *Communication protocols for vehicle cooperation:* Establishing reliable and low-latency communication protocols that facilitate the exchange of information between vehicles (V2V) and between vehicles and infrastructure (V2I). This ensures that cooperative driving decisions are based on the most current data.

3. *Impact assessment on traffic efficiency and safety:* Analyzing how multi-vehicle cooperative systems affect traffic flow, congestion levels, and accident rates. Simulation models and real-world experiments can help quantify these impacts and guide further development.

   Integrating multi-vehicle cooperative driving systems into the SMDT can lead to more harmonious traffic environments, reducing congestion and improving safety for all road users.

6.2.3 Causality in SMDT and World Model

Understanding and modeling causality within the SMDT and the associated world model is crucial for enhancing the decision-making capabilities of autonomous vehicles. By incorporating causal reasoning, the SMDT can provide more accurate predictions and support more effective planning and control strategies.

Future research directions include:

1. *Causal inference methods:* Applying statistical and machine learning techniques to infer causal relationships from the vast amounts of data collected by the SMDT. This involves distinguishing between correlation and causation to understand the true drivers of events within the transportation system.

2. *Integration of causal models in decision-making:* Embedding causal models into the autonomous vehicle's planning and control algorithms. This allows vehicles to anticipate the consequences of their actions and the actions of others, leading to more informed and safer decisions.

3. *Dynamic world modeling:* Developing world models that not only represent the current state of the environment but also predict future states based on causal relationships. This enhances the vehicle's ability to plan ahead and adapt to changing conditions.

By focusing on causality, future work can enhance the SMDT's ability to model complex interactions within the transportation system, leading to smarter and more explainable decision-making processes.

Appendix I

List of Publications

I.1 Journal papers

- **K. Wang**, Z. Li, K. Nonomura, T. Yu, K. Sakaguchi, O. Hashash, and W. Saad, "Smart Mobility Digital Twin Based Automated Vehicle Navigation System: A Proof of Concept," in *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 3, pp. 4348-4361, March 2024.

- **K. Wang**, K. Nonomura, Z. Li, T. Yu, K. Sakaguchi, O. Hashash, W. Saad, C. She, and Y. Li, "Augmented Intelligence in Smart Intersections: Local Digital Twins-Assisted Hybrid Autonomous Driving," in *IEEE Transactions on Intelligent Vehicles*, Early Access Article.

- Z. Li, **K. Wang**, T. Yu, and K. Sakaguchi, "Het-SDVN: SDN-Based Radio Resource Management of Heterogeneous V2X for Cooperative Perception," in *IEEE Access*, vol. 11, pp. 76255-76268, 2023.

I.2 International conferences

- **K. Wang**, Z. Li, T. Yu, and K. Sakaguchi, "Smart Mobility Digital Twin for Automated Driving: Design and Proof-of-Concept," *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, Florence, Italy, pp. 1-6, 2023.

- **K. Wang**, T. Yu, Z. Li, K. Sakaguchi, O. Hashash, and W. Saad, "Digital Twins for Autonomous Driving: A Comprehensive Implementation and Demonstration," *2024*

International Conference on Information Networking (ICOIN), Ho Chi Minh City, Vietnam, pp. 452-457, 2024.

- **K. Wang**, C. She, Z. Li, T. Yu, Y. Li, and K. Sakaguchi, "Roadside Units Assisted Localized Automated Vehicle Maneuvering: An Offline Reinforcement Learning Approach," *27th IEEE International Conference on Intelligent Transportation Systems (IEEE ITSC 2024)*, Edmonton, Canada, 2024.
- K. Nonomura, **K. Wang**, G. Cho, H. Matsuo, J. Nakazato, Z. Li, T. Yu, and K. Sakaguchi, "Proof-of-Concept of Digital Twin for Road Safety Assisted by B5G Heterogeneous MEC Network," *2024 IEEE 21st Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, pp. 1110-1111, 2024.

I.3 Domestic conferences

- **K. Wang**, Z. Li, T. Yu, and K. Sakaguchi, "Cloud and Edge Computing Empowered Mobility Digital Twin for Automated Driving: Design and Proof-of-Concept." *IEICE Technical Report*, vol. 123, no. 76, RCS2023-39, pp. 63-68, 2023.

Appendix II

Abbreviations

3GPP	Third Generation Partnership Project
ADS	Autonomous Driving System
AI	Artificial Intelligence
BC	Behavior Cloning
BEV	Birds-Eye-View
CAGR	Compound Annual Growth Rate
CAV	Connected and Automated Vehicles
CNN	Convolutional Neural Networks
CoR	Curse of Rarity
CPM	Cooperative Perception Messaging
CPS	Cyber-Physical System
CV	Connected Vehicle
C-V2X	Cellular V2X
DARPA	Defense Advanced Research Projects Agency
DL	Deep Learning
DL	Downlink
DNN	Deep Neural Networks
DQN	Deep Q-networks
DRL	Deep Reinforcement Learning
DSRC	Dedicated and Short-Range Communications
DT	Digital Twin
E2E	End-to-End

ECU	Electronic Control Unit
ETA	Estimated Time of Arrival
ETSI	European Telecommunications Standards Institute
FIFO	First-In-First-Out
GNN	Graph Neural Network
GNSS	Global Navigation Satellite System
GPU	Graphics Processing Unit
HD	High-Definition
HiL	Hardware-in-the-Loop
HTTP	Hypertext Transfer Protocol
I2N	Infrastructure-to-Network
IMU	Inertial Measurement Units
IoT	Internet of Things
ITE	Institute of Transportation Engineers
ITS	Intelligent Transportation System
KPI	Key Performance Indicator
LAN	Local Area Network
LDT	Local Digital Twin
LLVM	Large Language Vision Model
LQR	Linear Quadratic Regulators
LSTM	Long Short-Term Memory
LVM	Large Vision Model
MDP	Markov Decision Process
ML	Machine Learning
mMTC	massive Machine-Type Communications
mmWave	millimeter Wave
MOT	Multiple Object Tracking
MPC	Model Predictive Control
MTOS	Multi-Task Optimization Strategy
NASA	National Aeronautics and Space Administration
NDT	Normal Distributions Transform
NHTSA	National Highway Traffic Safety Administration
NLP	Nonlinear Programming

OBU	Onboard Unit
OSM	Open Street Map
PDR	Packet Delivery Rate
PID	Proportional-Integral-Derivative
PLM	Product Lifecycle Management
PoC	Proof-of-Concept
POMDP	Partially Observed Markov Decision Process
PPO	Proximal Policy Optimization
QoS	Quality of Service
RL	Reinforcement Learning
ROS	Robot Operating System
ROS2	Version 2 of Robot Operating System
RRT	Rapidly-exploring Random Trees
RSU	Roadside Unit
RTK	Real-Time Kinematic
SAE	Society of Automotive Engineers
SDN	Software-Defined Network
SDVN	Software-Defined Vehicular Network
SLAM	Simultaneous Localization and Mapping
SMDT	Smart Mobility Digital Twin
SOTA	State-Of-The-Art
SSMS	Sensor and State Map Sharing
TCP	Transmission Control Protocol
TD3+BC	Twin Delayed Deep Deterministic Policy Gradient with Behavior Cloning
TR	Technical Report
TTC	Time-To-Collision
UDP	User Datagram Protocol
UL	Uplink
URLLC	Ultra-Reliable Low-Latency Communications
V2I	Vehicle-to-Infrastructure
V2N	Vehicle-to-Network
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything

References

[1] C. Thorpe, M. Herbert, T. Kanade, and S. Shafer. Toward autonomous driving: the cmu navlab. i. perception. *IEEE Expert*, Vol. 6, No. 4, pp. 31–42, 1991.

[2] M. Williams. Prometheus-the european research programme for optimising the road transport system in europe. In *IEE Colloquium on Driver Information*, pp. 1/1–1/9, 1988.

[3] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*, Vol. 56. Springer Science & Business Media, 2009.

[4] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of field Robotics*, Vol. 25, No. 8, pp. 425–466, 2008.

[5] SAE On-Road Automated Vehicle Standards Committee, et al. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems J3016_202104. *SAE Standard Journal*, Vol. 3016, p. 4, 2021.

[6] Murat Dikmen and Catherine Burns. Trust in autonomous vehicles: The case of tesla autopilot and summon. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1093–1098, 2017.

[7] Neal E Boudette. 'it happened so fast': Inside a fatal tesla autopilot accident. *International New York Times*, pp. NA–NA, 2021.

[8] National Highway Traffic Safety Administration, et al. Automated driving systems 2.0: A vision for safety. *Washington, DC: US Department of Transportation, DOT HS*, Vol. 812, p. 442, 2017.

[9] Waymo One is now open to all in Los Angeles. `https://waymo.com/blog/2024/11/waymo-one-open-to-all-in-los-angeles/`. [Accessed 20-Nov-2024].

[10] Robotaxis - arriving at a future near you. `https://www.chinadailyhk.com/hk/article/593567/`. [Accessed 20-Nov-2024].

[11] Autonomous Vehicle Market Size, Share, and Trends 2024 to 2034. `https://www.precedenceresearch.com/autonomous-vehicle-market`. [Accessed 20-Nov-2024].

[12] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, Vol. 8, pp. 58443–58469, 2020.

[13] Rowan Thomas McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. In *International Joint Conferences on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence, Inc., 2017.

[14] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE international conference on computer vision*, pp. 2722–2730, 2015.

[15] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532*, 2017.

[16] Yaodong Cui, Shucheng Huang, Jiaming Zhong, Zhenan Liu, Yutong Wang, Chen Sun, Bai Li, Xiao Wang, and Amir Khajepour. Drivellm: Charting the path toward full autonomous driving with large language models. *IEEE Transactions on Intelligent Vehicles*, Vol. 9, No. 1, pp. 1450–1464, 2024.

[17] Henry X Liu and Shuo Feng. Curse of rarity for autonomous vehicles. *nature communications*, Vol. 15, No. 1, p. 4808, 2024.

[18] Zachary MacHardy, Ashiq Khan, Kazuaki Obana, and Shigeru Iwashina. V2x access technologies: Regulation, research, and remaining challenges. *IEEE Communications Surveys & Tutorials*, Vol. 20, No. 3, pp. 1858–1877, 2018.

[19] Zhengwei Bai, Guoyuan Wu, Matthew J. Barth, Yongkang Liu, Emrah Akin Sisbot, Kentaro Oguchi, and Zhitong Huang. A survey and framework of cooperative perception: From heterogeneous singleton to hierarchical cooperation. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 25, No. 11, pp. 15191–15209, 2024.

[20] Prasenjit Ghorai, Azim Eskandarian, Young-Keun Kim, and Goodarz Mehr. State estimation and motion prediction of vehicles and vulnerable road users for cooperative autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 10, pp. 16983–17002, 2022.

[21] ETSI. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions. *Technical Report 102 638 (V1.1.1)*, Jun. 2009.

[22] 3GPP. Study on enhancement of 3GPP support for 5G V2X services (Release 16). *Technical Report 22.886 (V16.2.0)*, Dec. 2018.

[23] Edward A Lee. Cyber physical systems: Design challenges. In *2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)*, pp. 363–369. IEEE, 2008.

[24] Jinghua Guo, Lubin Li, Jingyao Wang, and Keqiang Li. Cyber-physical system-based path tracking control of autonomous vehicles under cyber-attacks. *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 5, pp. 6624–6635, 2023.

[25] Armando W. Colombo, Stamatis Karnouskos, Okyay Kaynak, Yang Shi, and Shen Yin. Industrial cyberphysical systems: A backbone of the fourth industrial revolution. *IEEE Industrial Electronics Magazine*, Vol. 11, No. 1, pp. 6–16, 2017.

[26] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo. Cyber-physical systems security—a survey. *IEEE Internet of Things Journal*, Vol. 4, No. 6, pp. 1802–1831, 2017.

[27] Itxaro Errandonea, Sergio Beltrán, and Saioa Arrizabalaga. Digital twin for maintenance: A literature review. *Computers in Industry*, Vol. 123, p. 103316, 2020.

[28] Michael Grieves. Digital twin: Manufacturing excellence through virtual factory replication. 2014.

[29] Azure Digital Twins. https://azure.microsoft.com/en-us/products/digital-twins. [Accessed 20-Nov-2024].

[30] Siemens Xcelerator Digital Twins. https://xcelerator.siemens.com/global/en/all-offerings/products/digital-twin.html. [Accessed 20-Nov-2024].

[31] NVIDIA Omniverse Digital Twin Overview. https://docs.omniverse.nvidia.com/digital-twins/latest/index.html. [Accessed 20-Nov-2024].

[32] Tokyo Digital Twin Project. https://info.tokyo-digitaltwin.metro.tokyo.lg.jp/. [Accessed 20-Nov-2024].

[33] Andrey Rudskoy, Igor Ilin, and Andrey Prokhorov. Digital twins in the intelligent transport systems. *Transportation Research Procedia*, Vol. 54, pp. 927–935, 2021.

[34] Digital Twin Market Size, Share & Trends Analysis Report By End-use (Manufacturing, Agriculture), By Solution (Component, Process, System), By Region, And Segment Forecasts, 2024 - 2030. https://www.grandviewresearch.com/industry-analysis/digital-twin-market. [Accessed 20-Nov-2024].

[35] Ziran Wang, Rohit Gupta, Kyungtae Han, Haoxin Wang, Akila Ganlath, Nejib Ammar, and Prashant Tiwari. Mobility digital twin: Concept, architecture, case study, and future challenges. *IEEE Internet of Things Journal*, Vol. 9, No. 18, pp. 17452–17467, 2022.

[36] Yunsheng Ma, Runjia Du, Amr Abdelraouf, Kyungtae Han, Rohit Gupta, and Ziran Wang. Driver digital twin for online recognition of distracted driving behaviors. *IEEE Transactions on Intelligent Vehicles*, pp. 1–13, 2024.

[37] Zijin Wang, Ou Zheng, Liangding Li, Mohamed Abdel-Aty, Carolina Cruz-Neira, and Zubayer Islam. Towards next generation of pedestrian and connected vehicle in-the-loop research: A digital twin co-simulation framework. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 4, pp. 2674–2683, 2023.

[38] Wenwei Yue, Changle Li, Guoqiang Mao, Nan Cheng, and Di Zhou. Evolution of road traffic congestion control: A survey from perspective of sensing, communication, and computation. *China Communications*, Vol. 18, No. 12, pp. 151–177, 2021.

[39] Yuepeng Cui, Hao Xu, Jianqing Wu, Yuan Sun, and Junxuan Zhao. Automatic vehicle tracking with roadside LiDAR data for the connected-vehicles system. *IEEE Intelligent Systems*, Vol. 34, No. 3, pp. 44–51, 2019.

[40] Li Wang, Xinyu Zhang, Ziying Song, Jiangfeng Bi, Guoxin Zhang, Haiyue Wei, Liyao Tang, Lei Yang, Jun Li, Caiyan Jia, and Lijun Zhao. Multi-modal 3D object detection in autonomous driving: A survey and taxonomy. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 7, pp. 3781–3798, 2023.

[41] Thibault Degrande, Simon Van den Eynde, Frederic Vannieuwenborg, Didier Colle, and Sofie Verbrugge. C-ITS road-side unit deployment on highways with ITS road-side systems: A techno-economic approach. *IET Intelligent Transport Systems*, Vol. 15, No. 7, pp. 863–874, 2021.

[42] Viktor Tihanyi, András Rövid, Viktor Remeli, Zsolt Vincze, Mihály Csonthó, Zsombor Pethő, Mátyás Szalai, Balázs Varga, Aws Khalil, and Zsolt Szalay. Towards cooperative perception services for ITS: Digital twin in the automotive edge cloud. *Energies*, Vol. 14, No. 18, p. 5930, 2021.

[43] Yunpeng Guo, Kai Zou, Shengdong Chen, Feng Yuan, and Fang Yu. 3D digital twin of intelligent transportation system based on road-side sensing. In *Journal of Physics: Conference Series*, Vol. 2083, p. 032022. IOP Publishing, 2021.

[44] Javier Barrachina, Piedad Garrido, Manuel Fogue, Francisco J Martinez, Juan-Carlos Cano, Carlos T Calafate, and Pietro Manzoni. Road side unit deployment: A density-based approach. *IEEE Intelligent Transportation Systems Magazine*, Vol. 5, No. 3, pp. 30–39, 2013.

[45] Ozan K Tonguz and Wantanee Viriyasitavat. Cars as roadside units: a self-organizing network solution. *IEEE Communications Magazine*, Vol. 51, No. 12, pp. 112–120, 2013.

[46] Rahul Razdan, Mustafa İlhan Akbaş, Raivo Sell, Mauro Bellone, Mahesh Menase, and Mohsen Malayjerdi. Polyverif: An open-source environment for autonomous vehicle validation and verification research acceleration. *IEEE Access*, Vol. 11, pp. 28343–28354, 2023.

[47] Yuhang Liu, Boyi Sun, Yonglin Tian, Xingxia Wang, Yin Zhu, Rouxing Huai, and Yu Shen. Software-defined active LiDARs for autonomous driving: A parallel intelligence-based adaptive model. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 8, pp. 4047–4056, 2023.

[48] Tengchan Zeng, Omid Semiari, Mingzhe Chen, Walid Saad, and Mehdi Bennis. Federated learning on the road autonomous controller design for connected and autonomous vehicles. *IEEE Transactions on Wireless Communications*, Vol. 21, No. 12, pp. 10407–10423, 2022.

[49] Yuankai He, Baofu Wu, Zheng Dong, Jian Wan, and Weisong Shi. Towards C-V2X enabled collaborative autonomous driving. *IEEE Transactions on Vehicular Technology*, 2023.

[50] Omar Sami Oubbati, Mohammed Atiquzzaman, Pascal Lorenz, Abdullah Baz, and Hosam Alhakami. Search: An SDN-enabled approach for vehicle path-planning. *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 12, pp. 14523–14536, 2020.

[51] Tamás Wágner, Tamás Ormándi, Tamás Tettamanti, and István Varga. SPaT/MAP V2X communication between traffic light and vehicles and a realization with digital twin. *Computers and Electrical Engineering*, Vol. 106, p. 108560, 2023.

[52] Zongdian Li, Kui Wang, Tao Yu, and Kei Sakaguchi. Het-SDVN: SDN-based radio resource management of heterogeneous V2X for cooperative perception. *IEEE Access*, Vol. 11, pp. 76255–76268, 2023.

[53] Qiang Liu, Tao Han, Jiang Xie, and BaekGyu Kim. Real-time dynamic map with crowdsourcing vehicles in edge computing. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 4, pp. 2810–2820, 2023.

[54] Ziran Wang, Kyungtae Han, and Prashant Tiwari. Digital twin-assisted cooperative driving at non-signalized intersections. *IEEE Transactions on Intelligent Vehicles*, Vol. 7, No. 2, pp. 198–209, 2022.

[55] Jianghong Dong, Qing Xu, Jiawei Wang, Chunying Yang, Mengchi Cai, Chaoyi Chen, Yu Liu, Jianqiang Wang, and Keqiang Li. Mixed cloud control testbed: Validating vehicle-road-cloud integration via mixed digital twin. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 4, pp. 2723–2736, 2023.

[56] Kei Sakaguchi, Ryuichi Fukatsu, Tao Yu, Eisuke Fukuda, Kim Mahler, Robert Heath, Takeo Fujii, Kazuaki Takahashi, Alexey Khoryaev, Satoshi Nagata, et al. Towards mmwave v2x in 5g and beyond to support automated driving. *IEICE Transactions on Communications*, Vol. 104, No. 6, pp. 587–603, 2021.

[57] Cheng Chang, Jiawei Zhang, Kunpeng Zhang, Wenqin Zhong, Xinyu Peng, Shen Li, and Li Li. BEV-V2X: Cooperative birds-eye-view fusion and grid occupancy prediction via V2X-based data sharing. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 11, pp. 4498–4514, 2023.

[58] Md Mostafizur Rahman Komol, Mohammed Elhenawy, Mahmoud Masoud, Andry Rakotonirainy, Sebastien Glaser, Merle Wood, and David Alderson. Deep RNN based prediction of driver's intended movements at intersection using cooperative awareness messages. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 24, No. 7, pp. 6902–6921, 2023.

[59] Jiayao Tan, Fan Lyu, Linyan Li, Fuyuan Hu, Tingliang Feng, Fenglei Xu, Zhang Zhang, Rui Yao, and Liang Wang. Dynamic V2X perception from road-to-vehicle vision. *IEEE Transactions on Intelligent Vehicles*, pp. 1–14, 2024.

[60] Zilin Huang, Sikai Chen, Yuzhuang Pian, Zihao Sheng, Soyoung Ahn, and David A. Noyce. Toward C-V2X enabled connected transportation system: RSU-based cooperative localization framework for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2024.

[61] Mao Shan, Karan Narula, Yung Fei Wong, Stewart Worrall, Malik Khan, Paul Alexander, and Eduardo Nebot. Demonstrations of cooperative perception: Safety and robustness in connected and automated vehicle operations. *Sensors*, Vol. 21, No. 1, p. 200, 2020.

[62] Kunyang Cai, Ting Qu, Bingzhao Gao, and Hong Chen. Consensus-based distributed cooperative perception for connected and automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 24, No. 8, pp. 8188–8208, 2023.

[63] Zonglin Meng, Xin Xia, Runsheng Xu, Wei Liu, and Jiaqi Ma. HYDRO-3D: Hybrid object detection and tracking for cooperative perception using 3D LiDAR. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 8, pp. 4069–4080, 2023.

[64] Zhu Xiao, Jinmei Shu, Hongbo Jiang, Geyong Min, Hongyang Chen, and Zhu Han. Overcoming occlusions: Perception task-oriented information sharing in connected and autonomous vehicles. *IEEE Network*, Vol. 37, No. 4, pp. 224–229, 2023.

[65] Shunli Ren, Zixing Lei, Zi Wang, Mehrdad Dianati, Yafei Wang, Siheng Chen, and Wenjun Zhang. Interruption-aware cooperative perception for V2X communication-aided autonomous driving. *IEEE Transactions on Intelligent Vehicles*, Vol. 9, No. 4, pp. 4698–4714, 2024.

[66] Peng Wu, Xianbin Li, Hui Zheng, Kai Wang, Jiangyi Qin, and Min Tang. 3D modeling and analysis of cooperative perception-oriented millimeter-wave V2I networks with information value-based relay. *IEEE Transactions on Vehicular Technology*, Vol. 72, No. 8, pp. 10505–10520, 2023.

[67] Mohammad Sajid Shahriar, Arati Kantu Kale, and Kyunghi Chang. Enhancing intersection traffic safety utilizing V2I communications: Design and evaluation of machine learning based framework. *IEEE Access*, Vol. 11, pp. 106024–106036, 2023.

[68] Amit Chougule, Vinay Chamola, Vikas Hassija, Pranav Gupta, and Fei Richard Yu. A novel framework for traffic congestion management at intersections using federated learning and vertical partitioning. *IEEE Transactions on Consumer Electronics*, Vol. 70, No. 1, pp. 1725–1735, 2024.

[69] Kai-Fung Chu, Albert Y. S. Lam, and Victor O. K. Li. Traffic signal control using end-to-end off-policy deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 7, pp. 7184–7195, 2022.

[70] Yixiao Zhang, Rui Hao, Tingting Zhang, Xiaohan Chang, Zepeng Xie, and Qinyu Zhang. A trajectory optimization-based intersection coordination framework for cooperative autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 9, pp. 14674–14688, 2021.

[71] Xishun Liao, Ziran Wang, Xuanpeng Zhao, Kyungtae Han, Prashant Tiwari, Matthew J Barth, and Guoyuan Wu. Cooperative ramp merging design and field implementation: A digital twin approach based on vehicle-to-cloud communication. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 5, pp. 4490–4500, 2021.

[72] Annkathrin Krämmer, Christoph Schöller, Dhiraj Gulati, Venkatnarayanan Lakshminarasimhan, Franz Kurz, Dominik Rosenbaum, Claus Lenz, and Alois Knoll. Providentia–a large-scale sensor system for the assistance of autonomous vehicles and its evaluation. *arXiv preprint arXiv:1906.06789*, 2019.

[73] Zhen Yang, Yiheng Feng, and Henry X Liu. A cooperative driving framework for urban arterials in mixed traffic conditions. *Transportation research part C: emerging technologies*, Vol. 124, p. 102918, 2021.

[74] Dongwei Xu, Peiwen Liu, Haijian Li, Haifeng Guo, Zijia Xie, and Qi Xuan. Multi-view graph convolution network reinforcement learning for CAVs cooperative control in highway mixed traffic. *IEEE Transactions on Intelligent Vehicles*, Vol. 9, No. 1, pp. 2588–2599, 2024.

[75] Ziran Wang, Guoyuan Wu, and Matthew J Barth. Cooperative eco-driving at signalized intersections in a partially connected and automated vehicle environment. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 21, No. 5, pp. 2029–2038, 2019.

[76] Xingqi Wang, Chaoyang Jiang, Shuxuan Sheng, and Yanjie Xu. Stop-line-aided cooperative positioning for connected vehicles. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 2, pp. 1765–1776, 2023.

[77] Haoji Liu, Weichao Zhuang, Guodong Yin, Zhaojian Li, and Dongpu Cao. Safety-critical and flexible cooperative on-ramp merging control of connected and automated vehicles in mixed traffic. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 24, No. 3, pp. 2920–2934, 2023.

[78] Zakaria Charouh, Amal Ezzouhri, Mounir Ghogho, and Zouhair Guennoun. Video analysis and rule-based reasoning for driving maneuver classification at intersections. *IEEE Access*, Vol. 10, pp. 45102–45111, 2022.

[79] Jianda Wang, Zhendong Wang, Bo Yu, Jie Tang, Shuaiwen Leon Song, Cong Liu, and Yang Hu. Data fusion in infrastructure-augmented autonomous driving system: Why? where? and how? *IEEE Internet of Things Journal*, Vol. 10, No. 18, pp. 15857–15871, 2023.

[80] Muhammad Sami Irfan, Sagar Dasgupta, and Mizanur Rahman. Toward transportation digital twin systems for traffic safety and mobility: A review. *IEEE Internet of Things Journal*, Vol. 11, No. 14, pp. 24581–24603, 2024.

[81] Zhaoliang Zheng, Xu Han, Xin Xia, Letian Gao, Hao Xiang, and Jiaqi Ma. Opencdaros: Enabling seamless integration of simulation and real-world cooperative driving automation. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 7, pp. 3775–3780, 2023.

[82] Martin Buechel, Malte Schellmann, Holger Rosier, Tobias Kessler, and Alois Knoll. Fortuna: presenting the 5G-connected automated vehicle prototype of the project providentia. *Researchgate. https://doi. org/10.13140/RG*, Vol. 2, No. 24402.91842, 2019.

[83] Google Maps. `https://www.google.com/maps/`. [Accessed 20-Nov-2024].

[84] Baidu Maps. `https://map.baidu.com/`. [Accessed 20-Nov-2024].

[85] Mapbox Maps. `https://www.mapbox.com/maps`. [Accessed 20-Nov-2024].

[86] Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, et al. Eta prediction with graph neural networks in google maps. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 3767–3776, 2021.

[87] Jizhou Huang, Zhengjie Huang, Xiaomin Fang, Shikun Feng, Xuyi Chen, Jiaxiang Liu, Haitao Yuan, and Haifeng Wang. Dueta: Traffic congestion propagation pattern modeling via efficient graph learning for eta prediction at baidu maps. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 3172–3181, 2022.

[88] Hatem Darweesh, Eijiro Takeuchi, and Kazuya Takeda. Openplanner 2.0: The portable open source planner for autonomous driving applications. In *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pp. 313–318. IEEE, 2021.

[89] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 38, No. 10, pp. 2024–2039, 2015.

[90] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, Vol. 6, No. 3, pp. 4947–4954, 2021.

[91] Jianan Liu, Qiuchi Zhao, Weiyi Xiong, Tao Huang, Qing-Long Han, and Bing Zhu. Smurf: Spatial multi-representation fusion for 3d object detection with 4d imaging radar. *IEEE Transactions on Intelligent Vehicles*, Vol. 9, No. 1, pp. 799–812, 2024.

[92] How 10 leading companies are trying to make powerful, low-cost LiDAR. `https://arstechnica.com/cars/2019/02/the-ars-technica-guide-to-the-lidar-industry/`. [Accessed 20-Nov-2024].

[93] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11784–11793, 2021.

[94] J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[95] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024.

[96] Multi frame Lidar based detection with velocity (customized CenterPoint). `https://www.youtube.com/watch?v=-hpJg8q_shg`. [Accessed 20-Nov-2024].

[97] Congchao Wang, Yizhi Wang, Yinxue Wang, Chiung-Ting Wu, and Guoqiang Yu. mussp: Efficient min-cost flow algorithm for multi-object tracking. *Advances in neural information processing systems*, Vol. 32, , 2019.

[98] Jen-Hao Cheng, Sheng-Yao Kuan, Hou-I Liu, Hugo Latapie, Gaowen Liu, and Jenq-Neng Hwang. Centerradarnet: Joint 3d object detection and tracking framework using 4d fmcw radar. In *2024 IEEE International Conference on Image Processing (ICIP)*, pp. 998–1004. IEEE, 2024.

[99] Nicola Marinello, Marc Proesmans, and Luc Van Gool. Triplettrack: 3d object tracking using triplet embeddings and lstm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4500–4510, 2022.

[100] Jingkang Yang, Yuhao Dong, Shuai Liu, Bo Li, Ziyue Wang, Haoran Tan, Chencheng Jiang, Jiamu Kang, Yuanhan Zhang, Kaiyang Zhou, et al. Octopus: Embodied vision-language programmer from environmental feedback. In *European Conference on Computer Vision*, pp. 20–38. Springer, 2025.

[101] Jiachong Chang, Runzhi Hu, Feng Huang, Dingjie Xu, and Li-Ta Hsu. Lidar-based ndt matching performance evaluation for positioning in adverse weather conditions. *IEEE Sensors Journal*, 2023.

[102] Huan Yin, Runjian Chen, Yue Wang, and Rong Xiong. Rall: End-to-end radar localization on lidar map using differentiable measurement model. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 7, pp. 6737–6750, 2022.

[103] Shoubin Chen, Hao Ma, Changhui Jiang, Baoding Zhou, Weixing Xue, Zhenzhong Xiao, and Qingquan Li. Ndt-loam: A real-time lidar odometry and mapping with weighted ndt and lfa. *IEEE Sensors Journal*, Vol. 22, No. 4, pp. 3660–3671, 2021.

[104] Zongdian Li, Miguel Luis R. Lagahit, Masashi Matsuoka, and Kei Sakaguchi. Design of mobility-aware map partition and distribution system for smooth automated driving. In *2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 628–634, 2022.

[105] Benedict Flade, Simon Kohaut, and Julian Eggert. Error decomposition for hybrid localization systems. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 149–156. IEEE, 2021.

[106] Wei Liu, Jiaqi Zhu, Guirong Zhuo, Wufei Fu, Zonglin Meng, Yishi Lu, Min Hua, Feng Qiao, You Li, Yi He, et al. Unimsf: A unified multi-sensor fusion framework for intelligent transportation system global localization. *arXiv preprint arXiv:2409.12426*, 2024.

[107] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[108] Chen Feng, Hangning Zhou, Huadong Lin, Zhigang Zhang, Ziyao Xu, Chi Zhang, Boyu Zhou, and Shaojie Shen. Macformer: Map-agent coupled transformer for real-time and robust trajectory prediction. *IEEE Robotics and Automation Letters*, 2023.

[109] Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. *Advances in neural information processing systems*, Vol. 33, pp. 19783–19794, 2020.

[110] Yanchen Guan, Haicheng Liao, Zhenning Li, Jia Hu, Runze Yuan, Yunjian Li, Guohui Zhang, and Chengzhong Xu. World models for autonomous driving: An initial survey. *IEEE Transactions on Intelligent Vehicles*, pp. 1–17, 2024.

[111] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, 2023.

[112] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, Vol. 30, No. 7, pp. 846–894, 2011.

[113] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ international conference on intelligent robots and systems*, pp. 2997–3004. IEEE, 2014.

[114] Jonathan D Gammell, Timothy D Barfoot, and Siddhartha S Srinivasa. Batch informed trees (bit*): Informed asymptotically optimal anytime search. *The International Journal of Robotics Research*, Vol. 39, No. 5, pp. 543–567, 2020.

[115] Tim Mercy, Ruben Van Parys, and Goele Pipeleers. Spline-based motion planning for autonomous guided vehicles in a dynamic environment. *IEEE Transactions on Control Systems Technology*, Vol. 26, No. 6, pp. 2182–2189, 2017.

[116] Ling Zheng, Pengyun Zeng, Wei Yang, Yinong Li, and Zhenfei Zhan. Bézier curve-based trajectory planning for autonomous vehicles with collision avoidance. *IET Intelligent Transport Systems*, Vol. 14, No. 13, pp. 1882–1891, 2020.

[117] Sanghoon Oh, Qi Chen, H Eric Tseng, Gaurav Pandey, and Gábor Orosz. Sharable clothoid-based continuous motion planning for connected automated vehicles. *IEEE Transactions on Control Systems Technology*, 2024.

[118] Jie Ji, Amir Khajepour, Wael William Melek, and Yanjun Huang. Path planning and tracking for vehicle collision avoidance based on model predictive control with multi-constraints. *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 2, pp. 952–964, 2017.

[119] Hongyan Guo, Chen Shen, Hui Zhang, Hong Chen, and Rui Jia. Simultaneous trajectory planning and tracking using an mpc method for cyber-physical systems: A case study of obstacle avoidance for an intelligent vehicle. *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 9, pp. 4273–4283, 2018.

[120] Geraint P Bevan, H Gollee, and J O'reilly. Trajectory generation for road vehicle obstacle avoidance using convex optimization. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Vol. 224, No. 4, pp. 455–473, 2010.

[121] Kai Zhong, Prateek Jain, and Ashish Kapoor. Fast second-order cone programming for safe mission planning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 79–86. IEEE, 2017.

[122] Peide Cai, Yuxiang Sun, Yuying Chen, and Ming Liu. Vision-based trajectory planning via imitation learning for autonomous vehicles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2736–2742, 2019.

[123] Long Chen, Xuemin Hu, Bo Tang, and Yu Cheng. Conditional dqn-based motion planning with fuzzy logic for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 4, pp. 2966–2977, 2020.

[124] Dhruv Mauria Saxena, Sangjae Bae, Alireza Nakhaei, Kikuo Fujimura, and Maxim Likhachev. Driving in dense traffic with model-free reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5385–5392, 2020.

[125] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pp. 414–430. Springer, 2020.

[126] Omead Amidi and Chuck E Thorpe. Integrated mobile robot control. In *Mobile Robots V*, Vol. 1388, pp. 504–523. SPIE, 1991.

[127] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. *The 2005 DARPA grand challenge: the great robot race*, Vol. 36. Springer Science & Business Media, 2007.

[128] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*, Vol. 56. Springer Science & Business Media, 2009.

[129] Paolo Falcone, Francesco Borrelli, Jahan Asgari, Hongtei Eric Tseng, and Davor Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, Vol. 15, No. 3, pp. 566–580, 2007.

[130] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532*, 2017.

[131] Xing Fang, Qichao Zhang, Yinfeng Gao, and Dongbin Zhao. Offline reinforcement learning for autonomous driving with real world driving data. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3417–3422. IEEE, 2022.

[132] Changxi You, Jianbo Lu, Dimitar Filev, and Panagiotis Tsiotras. Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. *Robotics and Autonomous Systems*, Vol. 114, pp. 1–18, 2019.

[133] Amazon Web Services (AWS). `https://aws.amazon.com/`. [Accessed 20-Nov-2024].

[134] Google Cloud Platform (GCP). `https://cloud.google.com/`. [Accessed 20-Nov-2024].

[135] Apollo Cloud. `https://apollocloud.baidu.com/`. [Accessed 20-Nov-2024].

[136] Web Auto. `https://web.auto/`. [Accessed 20-Nov-2024].

[137] NVIDIA Drive Platform. `https://www.nvidia.com/en-us/self-driving-cars/`. [Accessed 20-Nov-2024].

[138] NVIDIA Jetson Platform. `https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/`. [Accessed 20-Nov-2024].

[139] Lenovo ThinkSystem SE350. `https://lenovopress.lenovo.com/lp1069-introducing-the-thinksystem-se350`. [Accessed 20-Nov-2024].

[140] Intel NUC. `https://www.intel.sg/content/www/xa/en/products/docs/boards-kits/nuc/mini-pcs/nuc-12-pro.html?wapkw=NUC`. [Accessed 20-Nov-2024].

[141] Raspberry Pi. `https://www.raspberrypi.com/`. [Accessed 20-Nov-2024].

[142] 3GPP. Study on traffic characteristics and performance requirements for AI/ML model transfer in 5GS (Release 18). *Technical Report 22.874 V18.2.0*, Dec. 2021.

[143] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. Digital twin in manufacturing: A categorical literature review and classification. *Ifac-PapersOnline*, Vol. 51, No. 11, pp. 1016–1022, 2018.

[144] Eric VanDerHorn and Sankaran Mahadevan. Digital twin: Generalization, characterization and implementation. *Decision support systems*, Vol. 145, p. 113524, 2021.

[145] Fei Tao, He Zhang, Ang Liu, and Andrew YC Nee. Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 4, pp. 2405–2415, 2018.

[146] Christo Kurisummoottil Thomas, Walid Saad, and Yong Xiao. Causal semantic communication for digital twins: A generalizable imitation learning approach. *IEEE Journal on Selected Areas in Information Theory*, Vol. 4, pp. 698–717, 2023.

[147] Omar Hashash, Christina Chaccour, Walid Saad, Tao Yu, Kei Sakaguchi, and Merouane Debbah. The seven worlds and experiences of the wireless metaverse: Challenges and opportunities. *IEEE Communications Magazine*, 2024.

[148] Tao Yu, Zongdian Li, Omar Hashash, Kei Sakaguchi, Walid Saad, and Mérouane Debbah. Internet of federated digital twins: Connecting twins beyond borders for society 5.0. *IEEE Internet of Things Magazine*, Vol. 7, No. 5, pp. 64–71, 2024.

[149] Shuangshuang Han, Fei-Yue Wang, Guiyang Luo, Lefei Li, and Fengzhong Qu. Parallel surfaces: Service-oriented V2X communications for autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 11, pp. 4536–4545, 2023.

[150] Junchao Yang, Feng Lin, Chinmay Chakraborty, Keping Yu, Zhiwei Guo, Anh-Tu Nguyen, and Joel J. P. C. Rodrigues. A parallel intelligence-driven resource scheduling scheme for digital twins-based intelligent vehicular systems. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 4, pp. 2770–2785, 2023.

[151] Laurens Hobert, Andreas Festag, Ignacio Llatser, Luciano Altomare, Filippo Visintainer, and Andras Kovacs. Enhancements of V2X communication in support of cooperative autonomous driving. *IEEE Communications Magazine*, Vol. 53, No. 12, pp. 64–70, 2015.

[152] Omar Hashash, Christina Chaccour, Walid Saad, Kei Sakaguchi, and Tao Yu. Towards a decentralized metaverse: Synchronized orchestration of digital twins and submetaverses. In *ICC 2023 - IEEE International Conference on Communications*, pp. 1905–1910, 2023.

[153] Xuemin Hu, Shen Li, Tingyu Huang, Bo Tang, Rouxing Huai, and Long Chen. How simulation helps autonomous driving: A survey of sim2real, digital twins, and parallel intelligence. *IEEE Transactions on Intelligent Vehicles*, Vol. 9, No. 1, pp. 593–612, 2024.

[154] Yilun Lin, Xiaoxiang Na, Ding Wang, Xingyuan Dai, and Fei-Yue Wang. Mobility 5.0: Smart logistics and transportation services in cyber-physical-social systems. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 6, pp. 3527–3532, 2023.

[155] Autoware Universe Documentation. `https://autowarefoundation.github.io/autoware.universe/main/`. [Accessed 20-Nov-2024].

[156] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.

[157] UC-win/Road simulator. `https://www.forum8.co.jp/english/uc-win/ucwin-road-e1.htm`. [Accessed 20-Nov-2024].

[158] Abd-Elhamid Taha and Najah AbuAli. Route planning considerations for autonomous vehicles. *IEEE Communications Magazine*, Vol. 56, No. 10, pp. 78–84, 2018.

[159] Younes Regragui and Najem Moussa. A real-time path planning for reducing vehicles traveling time in cooperative-intelligent transportation systems. *Simulation Modelling Practice and Theory*, Vol. 123, p. 102710, 2023.

[160] P. B. Sujit, Daniel E. Lucani, and Joao B. Sousa. Bridging cooperative sensing and route planning of autonomous vehicles. *IEEE Journal on Selected Areas in Communications*, Vol. 30, No. 5, pp. 912–922, 2012.

[161] Xiaolong Chen, Biao Xu, Xiaohui Qin, Yougang Bian, Manjiang Hu, and Ning Sun. Non-signalized intersection network management with connected and automated vehicles. *IEEE Access*, Vol. 8, pp. 122065–122077, 2020.

[162] Hani S Mahmassani. Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and spatial economics*, Vol. 1, pp. 267–292, 2001.

[163] Bin Ran and David Boyce. *Modeling dynamic transportation networks: an intelligent transportation system oriented approach*. Springer Science & Business Media, 2012.

[164] Jon D Fricker and Robert K Whitford. Fundamentals of transportation engineering. *A Multimodal Systems Approach. Inc. Upper Saddle River, New Jersey, USA*, 2004.

[165] Institute of Transportation Engineers. Technical Council Committee 4A-16. Determining vehicle change intervals: A proposed recommended practice. *Institute of Transportation Engineers*, 1985.

[166] Mario H. Castañeda Garcia, Alejandro Molina-Galan, Mate Boban, Javier Gozalvez, Baldomero Coll-Perales, Taylan Şahin, and Apostolos Kousaridas. A tutorial on 5G NR V2X communications. *IEEE Communications Surveys & Tutorials*, Vol. 23, No. 3, pp. 1972–2026, 2021.

[167] Andrés F Acosta, Jorge E Espinosa, and Jairo Espinosa. Traci4matlab: Enabling the integration of the sumo road traffic simulator and matlab® through a software re-engineering process. In *Modeling Mobility with Open Data: 2nd SUMO Conference 2014 Berlin, Germany, May 15-16, 2014*, pp. 155–170. Springer, 2015.

[168] Shaoshan Liu, Liangkai Liu, Jie Tang, Bo Yu, Yifan Wang, and Weisong Shi. Edge computing for autonomous driving: Opportunities and challenges. *Proceedings of the IEEE*, Vol. 107, No. 8, pp. 1697–1716, 2019.

[169] Vinay Chamola, Amit Chougule, Aishwarya Sam, Amir Hussain, and Fei Richard Yu. Overtaking mechanisms based on augmented intelligence for autonomous driving: Data sets, methods, and challenges. *IEEE Internet of Things Journal*, Vol. 11, No. 10, pp. 17911–17933, 2024.

[170] Chen Xu, Gang Wang, Mingzhu Wei, Ping Zhang, and Bei Peng. Intelligent transportation vehicle road collaboration and task scheduling based on deep learning in augmented internet of things. *IEEE Transactions on Vehicular Technology*, pp. 1–12, 2024.

[171] Jinghui Yuan and Mohamed Abdel-Aty. Approach-level real-time crash risk analysis for signalized intersections. *Accident Analysis & Prevention*, Vol. 119, pp. 274–289, 2018.

[172] Statistics on Intersection Accidents. `https://autoaccident.com/statistics-on-intersection-accidents.html`. [Accessed 20-Nov-2024].

[173] Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, and Long Chen. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*, Vol. 8, No. 6, pp. 3692–3711, 2023.

[174] Ricardo Eiji Kondo, Willian Jeferson Andrade, Clayton de Mello Henequim, André Eugenio Lazzaretti, Alceu de Souza Britto Junior, Eduardo de Freitas Rocha Loures, Eduardo Alves Portela Santos, and Gilberto Reynoso-Meza. An industrial edge computing architecture for local digital twin. *Computers & Industrial Engineering*, p. 110257, 2024.

[175] Tarik Taleb, Chafika Benzaïd, Rami Akrem Addad, and Konstantinos Samdanis. AI/ML for beyond 5G systems: Concepts, technology enablers & solutions. *Computer Networks*, Vol. 237, p. 110044, 2023.

[176] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[177] Zenan Li, Fan Nie, Qiao Sun, Fang Da, and Hang Zhao. Boosting offline reinforcement learning for autonomous driving with hierarchical latent skills. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 18362–18369, 2024.

[178] Japanese Traffic Regulations for SOFA Licensed Drivers. `https://www.mcipac.marines.mil/Portals/28/Documents/Safety/Driver's%20Study%20Guide%202020(Updated%20on%20Aug17).pdf?ver=2020-08-25-021235-977`. [Accessed 20-Nov-2024].

[179] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, Vol. 34, pp. 20132–20145, 2021.

[180] Yonghua Huang, Qizheng Liao, Lei Guo, and Shimin Wei. Balanced motions realization for a mechanical regulators free and front-wheel drive bicycle robot under zero forward speed. *International Journal of Advanced Robotic Systems*, Vol. 10, No. 8, p. 317, 2013.