Northumbria University
NEWCASTLE

Department of Computer and Information Sciences

Deep Learning for Social-Aware Autonomous Vehicle-Pedestrian Interaction

Luca Crosato

This thesis is submitted for the degree of
Doctor of Philosophy

Year 2023

Abstract

In recent years, Autonomous Driving technology has surged in popularity, becoming a key research focus. Despite commercial advancements, many legal and technical challenges remain, particularly in AVs' interactions with human road users.

Motion control algorithms for AVs in pedestrian scenarios are crucial for safety and reliability. Traditional algorithms, which rely on manually designed policies, scale poorly with complexity and are costly. In contrast, Deep Reinforcement Learning (DRL) allows for automatic policy learning. This thesis explores automated AV decision-making in AV-pedestrian interactions using DRL and social psychology.Firstly, we propose a framework based on Social Value Orientation and Deep Reinforcement Learning capable of generating decision-making policies for the AV with different driving styles. Adding a social term in the reward function design allows us to tune the AV attitude towards the pedestrian from a more aggressive to an extremely prudent one. This model achieves a 0% collision rate and exhibits 10 behavioral modes.

We also introduce a novel pedestrian model incorporating situational awareness into a Social Force Model, enabling realistic pedestrian reactions to AV actions. We perform experiments to validate our framework and we conduct a comparative analysis of the policies obtained with two different model-free Deep Reinforcement Learning Algorithms. Comparative analysis of policies from two DRL algorithms, SAC and PPO, reveals that SAC-trained vehicles stop 30% earlier and maintain a 1.5-meter larger distance from pedestrians, while PPO policies yield 20% smoother acceleration profiles. Extending to multi-agent settings, we employ Graph Convolutional Networks to manage multiple vehicles and pedestrians, capturing agent inter-relationships efficiently.

Lastly, we develop a 3D Virtual Reality environment for studying pedestrian interactions with vehicles. Using VR technology, we collect data safely and cost-effectively. Graph neural networks predict pedestrian trajectories with a 0.17 m average displacement error, demonstrating the framework's effectiveness in studying pedestrian-vehicle interactions.

Keywords: Autonomous Driving, Deep Reinforcement Learning, Social Value Orientation, Pedestrian Modelling, Situational Awareness, Virtual Reality, Human-Robot Interaction.

List of Abbreviations

AD	Autonomous Driving
ADE	Average Displacement Error
AI	Artificial Intelligence
AV	Autonomous Vehicle
BC	Bimodal Crossing (behaviour)
BEV	Bird's Eye View (image)
CA	Cellular Automata
CIT	Crossing Initiation Time
DBM	Driver Behavioural Model
DRF	Driver Risk Field
DRL	Deep Reinforcement Learning
DQN	Deep Q-Learning
EA	Evidence Accumulation
FDE	Final Displacement Error
GA	Gap Acceptance
GAN	Generative Adversarial Network
GAT	Graph Attention Network
GCN	Graph Convolutional Network
GNN	Graph Neural Network
HDV	Human Driven Vehicle
HMD	Head Mounted Display
HMI	Human-Machine Interface
eHMI	External Human-Machine Interface
HMM	Hidden Markov Model
HRU	Human Road Users

IBR	Iterative Best Response
IDM	Intelligent Driver Model
IL	Imitation Learning
IRL	Inverse Reinforcement Learning
LQ	Linear Quadratic
LR	Logistic Regression
LSTM	Long Short-Term Memory
MARL	Multi-Agent Reinforcement Learning
MDP	Markov Decision Process
ML	Machine Learning
MLP	Multilayer Perceptron
MPC	Model Predictive Control
NN	Neural Network
PER	Prioritized Experience Replay
PGIM	Parallel-Game Interaction Model
POMDP	Partially Observable Markov Decision-Process
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RQ	Research Question
SAC	Soft Actor-Critic
SFM	Social Force Models
STGCNN	Spatio-Temporal Graph Convolutional Network
SVO	Social Value Orientation
SVM	Support Vector Machine
TD	Temporal Difference (error)
TTC	Time To Collision
UE	Unreal Engine
VR	Virtual Reality
VRU	Vulnerable Road User

Acknowledgements

I would like to take this opportunity to express my deepest gratitude and appreciation to the people who have helped me throughout this PhD journey.

First and foremost, I would like to thank my family and friends who supported me and helped me overcome all the struggles that I faced during this journey. A special thanks to my husband, Davide, my mother, Giulia, and Giacomo. Thank you for listening, offering me advice, and being the best supporters and vital parts of my life.

A special thanks to my principal supervisor, Dr. Chongfeng Wei. I am truly grateful for your guidance, support, and belief in my abilities. I would also like to thank the members of my PhD committee, Dr. Hubert P. H. Shum and Dr. Edmond S. L. Ho. I have deep gratitude for both of you for teaching me so much, for your invaluable feedback, and also for your emotional support during my toughest times throughout this PhD.

Finally, I want to acknowledge the countless individuals who, directly or indirectly, have contributed to this thesis. Your contributions may not be explicitly mentioned here, but you have all played a significant role in my academic growth and the completion of this work.

Author,
Luca Crosato

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Luca Crosato
September 2023

Publications

The following publications serve as the basis of this thesis. The author of the thesis was the main contributor to Paper 1-4. Paper 5 was written in collaboration with Dr Kai Tian and both authors contributed equally to this work. These papers have either been published or are under revision.

1. **L Crosato**, HPH Shum, ESL Ho, C Wei (2022). *Interaction-aware decision-making for automated vehicles using social value orientation.* In IEEE Transactions on Intelligent Vehicles, 8(2), 1339-1349.

2. **L Crosato**, C Wei, ESL Ho, HPH Shum (2021, September). *Human-centric autonomous driving in an av-pedestrian interactive environment using SVO.* In 2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS) (pp. 1-6). −**Best Paper Award**−.

3. **L Crosato**, HPH Shum, ESL Ho, C Wei (2023, September). *A Virtual Reality Framework for Human-Driver Interaction Research: Safe and Cost-Effective Data Collection.* Under Review.

4. **L Crosato**, HPH Shum, ESL Ho, C Wei (2023, September). *Game-Theoretic Strategies for Autonomous Vehicle Decision-Making: A Multi-Agent Study with Deep Learning and Graph Convolutional Networks.* Under Review.

5. **L Crosato**, K Tian, ESL Ho, HPH Shum, C Wei (2023, September). *Interaction-Aware Dynamical Models and Motion Planning for Autonomous Vehicles Interacting with Pedestrians: A Survey.* Under Review.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the past few years, there has been a growing interest in the development of technology for autonomous vehicles (AVs). This is due to recent advances in robotics and machine learning, which have enabled autonomous driving (AD) engineers to develop algorithms that can tackle the complexity of the autonomous driving task [1]. AVs have the potential to improve traffic quality, reduce traffic accidents, and improve the quality of time spent during travel [1]. Even though we have witnessed a rapid spread of Advanced Driver-Assistance Systems (ADAS), the number of deaths on the world's roads remains unacceptably high, with an estimated 1.35 million people dying each year, as stated in the Global Status Report on Road Safety [2]. Over the past few years, the automotive industry has shown an increasing interest in the development of autonomous driving, as ADAS have become a reality. ADAS use automated technology, such as LiDAR sensors and cameras, to detect nearby obstacles or driver errors, and respond accordingly. The level of automation of autonomous vehicles is standardized in the J3016 SAE document [3]. SAE defines six levels of automation, with a decreasing degree of human intervention. Level 0 represents vehicles with no automation, whereas level 5 represents a fully automated vehicle, capable of navigating through any kind of road scenario. A high degree of automation will require achieving high safety standards and overcoming, not only technical challenges, but also legal and ethical ones [4].

In order to operate in an efficient and safe manner, AVs need to behave in a human-like fashion and generate optimal behaviours that take the interactions with other agents into account [4]. This is critical for the reduction of potential traffic accidents, as unusual or unpredictable behaviour could negatively impact human driving safety. For example, cautious but unnecessary stops at intersections could cause rear-end collisions. Indeed, one of the current challenges in Autonomous Driving involves unconventional methods of communication (e.g. external Human Machine Interfaces) or movement patterns employed by AVs, such as unpredictable lane changes or sudden stops for pedestrians, which can potentially bewilder human drivers. This not only constitutes a potential danger during interactions between vehicles but can also result in a reluctance among the public to accept AVs. Inevitably, AVs will have to share

roads with Human Driven Vehicles (HDUs), originating mixed traffic scenarios with both AVs and humans (drivers, cyclists or pedestrians). Thus, it is imperative that AVs behave in a manner that mimics human interactions, as human drivers desire to continue their established practices for vehicle-to-vehicle communication [5].

Advances in many aspects of AV technology are required to develop fully automated vehicles,, ranging from perception, decision-making, planning, and control [6], [7]. When it comes to predicting the behaviour of surrounding traffic participants and taking decisions accordingly for AVs, the interactions with surrounding traffic participants become increasingly important, as the AV's actions affect the surrounding agents' behaviour and vice versa [8].

The main research area of this thesis is decision-making for AVs. The decision-making of an AV can be divided into three main components: strategical, tactical, and operational [9]. The strategical decision-making refers to the planning of a global route that handles the transportation mission. The tactical decision-making modifies the strategic level in response to the current traffic conditions. This would involve decisions such as whether to yield or cross at an intersection, whether or not it is suitable to change lane at a given time, and so on. Finally, the operational level translates the tactical level into lower-level commands (such as acceleration and steering) and defines a detailed trajectory. Low-level control of the vehicle is a mature research area and can be solved with classical control theory methods [7].

The focus of this thesis is interaction-aware autonomous driving in the presence of pedestrians. Interaction-aware autonomous driving is a type of autonomous driving that takes into account the interactions between the autonomous vehicle and other road users, such as other vehicles, pedestrians, and cyclists. This means that interaction-aware autonomous vehicles are designed to predict the behavior of other road users and to avoid collisions. This thesis addresses high-level tactical decision-making when pedestrians are present, along with a brief examination of predicting the movements of pedestrians around the ego-vehicle.

1.1 Approach

Numerous research articles have shifted their attention to the development of interaction-aware control approaches for AVs. These approaches include game theory [10], [11], Deep Reinforcement Learning [12], [13] and Model Predictive Control [14]. However, most of these methods are tailored for scenarios comprising AVs and HDUs, with less focus on pedestrian-AV interactions. Indeed, although autonomous driving has been an active research area in recent years, most literature focuses on scenarios involving only vehicles with more limited work that addresses heterogeneous scenarios, which include both vehicles and pedestrians [15]. Therefore, the first motivation of this thesis is the design of AV decision-making algorithms in the presence of **pedestrians**.

Current AV projects are aiming at an SAE level 4 or higher, i.e. at achieving fully

autonomous vehicles. This requires algorithms that are capable of handling complex situations. Many recent studies on autonomous vehicles control make use of sophisticated Artificial Intelligence (AI) algorithms that are capable of learning complex control policies without being explicitly handcrafted. In particular, recent advances in machine learning enable the possibility of learning-based approaches for autonomous driving decision-making. Combined with deep learning techniques, Reinforcement Learning (RL) is a very promising field which has achieved relevant breakthroughs in the last few years. Agents trained with RL have been capable of exceeding human-level performance in video games [16], [17], in the game of Go [18], in continuous control problems [19], and robotics [20], [21].

Besides, traditional learning-based methods (e.g. linear regression, decision trees), game theoretic models or optimal control do not scale well with increasing traffic dimensions and are mostly tailored for limited driving scenarios. Deep learning techniques have played a significant role in greatly enhancing perception systems [22] and have more recently been investigated in the decision-making domain [23]. Therefore, this thesis will study how DRL methods can be applied to interaction-aware decision-making for AV in the presence of pedestrians.

A challenge that arose during this research is that most DRL papers treat pedestrians as mere moving obstacles. To circumvent the introduction of a real human in the loop that would significantly slow down the DRL training process, it becomes essential to incorporate pedestrian models in simulations that mimic the behaviour of humans [23]. There is scarce literature of pedestrian models that deal with scenarios comprising multiple vehicles and lanes. Firstly, a comprehensive literature review has been conducted to find out relevant research questions that had to be addressed.

Identifying Research Gaps in Interaction-Aware AV Motion Planning and Decision-Making – Publication 5

Interaction-Aware Motion Planning and Decision-Making refers to the processes by which AVs navigate and make decisions while taking into account the interactions with other vehicles, pedestrians, and various elements in their environment. Identifying what aspects of Interaction-Aware AV Motion Planning and Decision-Making have not been extensively studied or remain unclear can help guide future research efforts and ultimately contribute to the development of safer and more effective autonomous vehicle systems. A comprehensive review of the existing literature pertaining to Interaction-Aware Decision-Making for Autonomous Vehicles (AVs) has been conducted. This review involved the analysis of more than 300 scholarly papers dedicated to this subject matter, and the findings have been synthesized into a survey paper, which is a component of publication 5 and part of Chapter 3 of this thesis. The outcome of this review has been instrumental in pinpointing several critical research gaps, offering valuable insights that serve as guiding principles for the research delineated in this thesis.

In particular, most of the existing research focuses on planning and decision making from AV from a robotics perspective, disregarding the role of social factors in

the decision-making process. Although some efforts have been made to include Social Psychology concepts into Game Theoretic motion planning methods, little to few studies focus on integrating such concepts into learning-based systems. We identify Deep Learning applications to Autonomous Driving system as a hot research topic in the area but little to no studies consider social aspects to aid decision-making. The integration of these essential social factors into the AV decision-making process forms the core of the responses to the research questions posed under RQ1, RQ2, RQ3, and RQ4.

In particular, the following research questions (RQs) are considered:

RQ1: How can we incorporate Social Psychology aspects to aid AV decision-making? – Publications 2

The potential integration of insights and principles from the field of Social Psychology into the process of decision-making for autonomous vehicles (AVs) is a under-researched topic. We are interested in exploring the potential benefits of using insights from Social Psychology to make autonomous vehicles better at understanding and responding to the social aspects of driving, including interactions with human drivers, pedestrians, and other social elements on the road. This could involve designing AV algorithms and systems that take into account human behavior, emotions, and social cues to make more contextually aware and socially responsible decisions.

Proposed Solution: In this thesis, we propose to exploit recent advancements in Deep Reinforcement Learning (DRL) to tackle this question. As mentioned in RQ1, DRL We introduce a framework based on Social Value Orientation and DRL that is capable of generating decision-making policies with different driving styles . Social Value Orientation (SVO) is a concept from social psychology and economics that refers to an individual's preferences or orientation regarding the distribution of resources or outcomes in social situations Social Value Orientation (SVO) describes an individual's predisposition or attitude toward the distribution of resources between themselves and others. It reflects how a person views fairness, cooperation, and self-interest in social interactions. IN particular, the addition of a social term allows us to tune the AV behaviour towards the pedestrian from a more reckless to an extremely prudent one. The ego-vehicle agent is trained with state of the art RL algorithms and it is shown that Social Value Orientation is an effective tool to obtain pro-social AV behaviour.

In this first study, a pedestrian-AV simulator is developed where we consider a typical straight road scenario with a single pedestrian. The pedestrian motion and response to vehicle is obtained via Social-Force based models from [24]. Although we are able to demonstrate that the introduction of SVO to the DRL framework provides some benefits, there are still a few limitations to this work. Firstly, the pedestrian model's gap acceptance curve is not similar to real world data. Secondly, the study focuses on single agent scenarios, so an immediate issue is how to extend the results to multi-agent scenarios and whether the social psychology aspects can still benefit

AV decision-making in more cluttered scenarios. We address this issues in RQ3 and RQ4.

Thesis chapters: the research question will be addressed in Chapter 4.

RQ2: What pedestrian models are suitable for DRL training? Publication – 1 Pedestrians are considered part of the environment during DRL training. The models that guide their behaviour must therefore be computationally efficient so as not to cause any bottleneck during training and drastically increase training time.

Proposed Solution: A novel computationally-efficient pedestrian model that combines the concepts of situational awareness [25] and Social-Force [26] is developed to determine the pedestrian trajectory under the vehicle influence. The ego-vehicle motion affects the pedestrian decisions by indirectly altering the available time-gap to complete crossing and the social forces acting on the pedestrian. In turn, pedestrian motion serves as a cue for the ego-vehicle controller, thereby mutually influencing each other. We evaluate our pedestrian model using a set of typical road scenarios and by comparing pedestrian motion statistics with real world data and state-of-the-art pedestrian models.

Then, agents trained with model-free DRL algorithms learn the interaction patterns with the pedestrian and exploit them to indirectly affect pedestrian motion. For instance, the vehicle learns the effect that its own acceleration on pedestrian's decisions, thereby hindering or favouring the pedestrian crossing. We demonstrate how our reward choice produces controllers that naturally exhibit human-like behaviour, with a plethora of different driving styles, ranging across a spectrum from aggressive to pro-social according to the choice of the SVO value. We conduct a set of qualitative and quantitative experiments aimed at evaluating the effect of SVO addition, and model performances under both nominal and high-risk scenarios.

Thesis chapters: the research question will be addressed in Chapter 4.

RQ3: How can cluttered multi-agent scenarios be tackled and what are the effects of integrating concepts from Social Psychology? – Publication 4 Autonomous Vehicles will need to operate in scenarios where multiple surrounding agents (other vehicles and pedestrians) are present. In the first part of the thesis (RQ2-3) the focus is mainly on single AV-driver interactions. Here we are looking on how to expand the developed methods to tackle a more complicated road scenario that includes multiple agents surrounding the ego-vehicle.

Proposed Solution: We have developed a multi-agent simulator, building upon the CARLO simulator [27]. This simulator encompasses various vehicles and pedestrians and serves as the foundational environment for training Deep Reinforcement Learning (DRL) agents.

The dynamic nature of traffic environments poses challenges for DRL when dealing with neural network structures featuring fixed input sizes. To address this issue,

we propose the utilization of state-of-the-art Graph Convolutional Network (GCN) architectures. Specifically, we introduce a neural network design that maintains invariance to the presentation order of surrounding traffic participants. This type of network boasts several advantages over conventional Multilayer Perceptron (MLP) or Convolutional Neural Network (CNN) architectures. Firstly, it can accept a high-level representation of the ego-agent's surroundings as input. Secondly, its input size is adaptable, making it suitable for applications like autonomous driving, where the number of surrounding agents can vary.

One of the prominent challenges in our simulator's development pertained to modeling pedestrian behavior. In cases where only a single vehicle was present near a pedestrian, existing literature offers numerous pedestrian models to address the issue. However, as the environment becomes more crowded, no pedestrian models were available in the literature to tackle complex scenarios, such as multi-lane crossings and interactions with multiple vehicles. Therefore, our study also proposes the development of a pedestrian model based on DRL. The pedestrian is trained using Level-k Game Theory and is compared against existing models in the scenarios in which those models are valid.

Furthermore, we investigate the effects of Social Value Orientation (SVO) in multi-agent scenarios to analyze its benefits.

Thesis chapters: the research question will be addressed in Chapter 5.

RQ4: How can real-world AV-Pedestrian interaction data be captured in a safe and cost effective way for evaluating the performance of DRL models? There is currently no complete theory that explains how human road users interact with vehicles, and studying them in real-world settings is often unsafe and time-consuming. This raises the research question of how to obtain such data in a safe and cost-effective way.

Proposed Solution: This study proposes a 3D Virtual Reality (VR) framework for studying how pedestrians interact with human-driven vehicles and autonomous vehicles. A Virtual Reality environment for single pedestrian-driver/AV interactions has been developed using Unreal Engine. Pedestrians can connect to the Virtual Reality environemnt via a VR headset and their posture is captured by a motion capture system and streamed directly inside the virtual reality environment. A driver is also connected to the VR environment, which creates a suitable environment in which they can safely interact with each other. The main advantages of VR data collection over real world data collection include: reduced costs, it is easy to design new scenarios, increased safety.

The proposed framework uses VR technology to collect data in a safe and cost-effective way, and deep learning methods are used to predict pedestrian trajectories. Graph neural networks have been used to model pedestrian future trajectories and probability of crossing the road. The results of this study show that the proposed framework can be used to collect high-quality data on pedestrian-vehicle interactions

in a safe and efficient manner. The data can then be used to develop new theories of human-vehicle interaction and to train autonomous vehicles to better interact with pedestrians. In particular, we are further looking to utilise the system to validate the methods proposed in the previous research questions to aid the development of AV research. The VR framework can be used to assess the human-likeness of algorithms before being deployed to the real world, in a cost effective fashion.

Thesis chapters: the research question will be addressed in Chapter 6.

1.2 Contributions

The main contributions of this thesis are:

- a novel conceptual framework to integrate social psychology into the AV design. In particular, the introduction of SVO into the DRL framwork to influence the ego-vehicle strategies, achieving behaviours that range from egoistic to pro-social, without affecting pedestrian safety;
- the introduction of a novel pedestrian simulation model that combines gap-acceptance methods with Social Force Models to model the pedestrian crossing behaviour;
- an extension of the aforementioned SVO introduction to multi-agent scenarios. Including a permutation-invariant neural network to tackle multi-agent scenarios;
- the development of a traffic simulator, with a wireless HMD device (HTC Vive) that allows the users freedom of movement in combination with a motion capture system;
- a framework where pedestrians and user controlled vehicles can coexist with each other, as well as with Autonomous Vehicles. This framework can be used in the future to aid Autonomous Driving research for validation and testing.

1.3 Limitations

In this section we highlight the limitations and assumptions of this thesis:

- The presented algorithms have been developed and tested in simulation environments. How these algorithms can be deployed into the real world is an interesting research question that falls beyond the scope of this thesis.
- The question of whether safety can be assured solely through a learning-based approach remains unresolved and falls beyond the scope of this thesis. Although

the collision rates and success rates are analysed, it is presumed that in real world applications there will always be an essential safety layer overseeing the decisions made by RL-based agents.

- The DRL framework assumes that the state of the environment is known to the agent. How an AV can produce a high level representation of the state space from raw sensor data (LiDAR and cameras) is also beyond the scope of this thesis. In this thesis it is assumed that a high-level representation of the environment surrounding the ego-vehicle is already available to the decision-making.

1.4 Thesis Outline

Chapter 2 introduces the technical background which is mostly relevant to the thesis.

Chapter 3 covers a vast amount of prior work to further motivate the problem and explore past studies.

Chapter 4 exhibits the benefits of using Social Psychology aspects in the design of AV policies in a single-vehicle single-pedestrian scenario. A 2D pedestrian simulator is developed and tests are performed to evaluate the AV model.

Chapter 5 extends the results of Chapter 4 to multi-agent scenarios in more complicated road layouts. A GCN neural network is constructed and trained with DRL. Additional tests are performed to evaluate the impact of SVO in multi-agent settings.

Chapter 6 introduces a VR framework that can be used to validate AV policies in a safe and cost effective way.

Chapter 2

Technical Background

This Chapter provides a summary of the most important concepts and introduces the notation that will be used in subsequent Chapters. Markod Decision Processes and Reinforcement learning, which will be used extensively in Chapters 4 and 5 are covered in Sections 2.1 and 2.2.

2.1 Markov Decision Processes

A Markov Decision Process (MDP) 2.1 is a discrete-time stochastic process that provides a framework to describe situations where the outcome of a process is partly under control of a decision-making agent and partly random. An MDP is represented by a 4-tuple $(S, A, \mathcal{T}, R)$:

- S is the set of states of the environment called the *state space*;
- A is the set of actions that the agent can take (*action space*);
- $\mathcal{T}(s_{t+1}|s_t, a_t)$ is the state transition probability, which indicates the probability of the environment evolving to state s_{t+1} from state s_t after the agent takes action a_t;
- and $R(s_{t+1}|s_t, a_t)$ is the immediate reward function received after transitioning to state s_{t+1} from state s_t with action a_t.

In an MDP, the agent (see Figure 2.2), which in Autonomous Driving can represent the ego-vehicle, interacts with its environment at discrete time steps. The agent performs an action $a_t \in A$ which causes a change in the environment state $s_t \in S$. The action taken is chosen according to a policy $\pi(a|s)$, which is a function that maps the current state to an action. $\pi(a|s)$ is mostly chosen as a stochastic function rather than a deterministic function. In turn, the environment gives the agent a numerical reward r_t and evolves to a new state $s_{t+1} \sim \mathcal{T}(s_{t+1}|s_t, a_t)$ sampled from the transition probability.

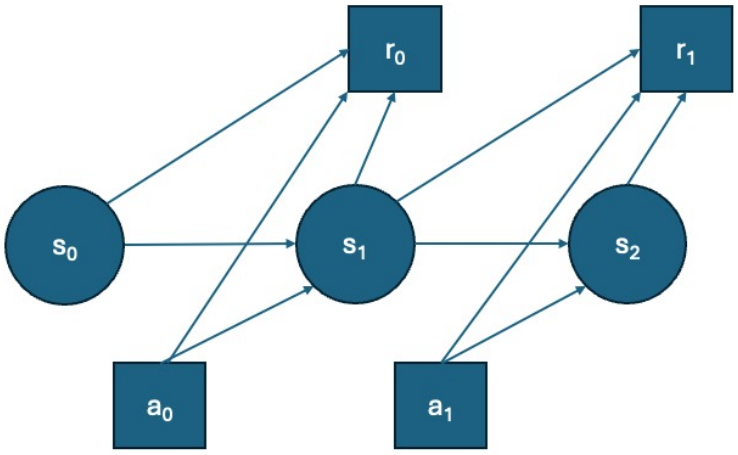


Figure 2.1: A Markov Decision Process with states, actions and rewards.

2.2 Reinforcement Learning

Reinforcement Learning (RL) is a branch of machine learning that is concerned with the study of decision-making of intelligent agents. RL uses the MDP framework to model the decision process and provides a set of algorithms and techniques to find optimal policies.

In many cases, the transition probability function $\mathcal{T}(s_{t+1}|s_t, a_t)$ is either unknown or difficult to model. A possible solution to overcome this problem is to use a simulator which provides samples from the transition distribution. The episodic environment simulator can be initiated from an initial state and provides a subsequent state and reward each time it is given an action input. This approach enables the generation of sequences of states, actions, and rewards, typically referred to as episodes. Reinforcement Learning can solve Markov-Decision processes with unknown transition probabilities. The goal of an RL algorithm is to learn an optimal policy $\pi^*(a|s)$, which is a mapping from the state space to the action space, that maximises the expected future total reward:

$$J(\pi) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k}] \tag{2.1}$$

where γ is called the discount factor. Gamma is less than 1, so events in the distant future are weighted less than events in the immediate future.

Reinforcement Learning can be combined with function approximators (e.g. Neural Networks) to address problems with large state spaces. When Neural Networks are combined with Reinforcement Learning it is named Deep Reinforcement Learning (DRL). In the following Subsections, we will go through some DRL algorithms that have been used throughout this thesis. Autonomous Driving has inherently large

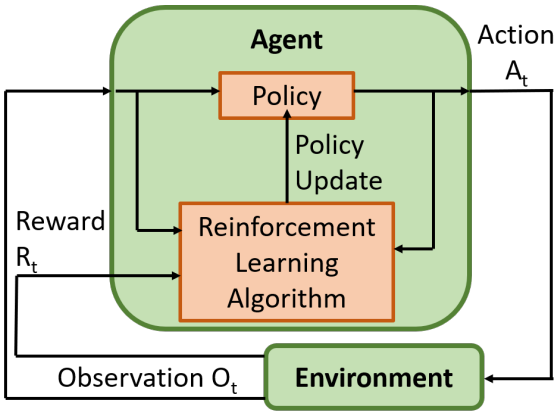


Figure 2.2: The RL framework. As the agent interacts with the environment, the RL algorithm updates the policy function based on the experience gathered in order to improve the policy and achieve better cumulative reward in the future.

state-spaces, which makes DRL suitable for the area.

2.2.1 Double Q-Learning

Q-learning is a model-free reinforcement learning algorithm used to learn the optimal action-selection policy for an agent in an environment. Q-Learning is an RL algorithm that seeks to find an optimal policy by estimating the so-called Q function, which is defined as the expected future cumulative reward under policy π after taking action a in state s:

$$Q(s,a) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a] \tag{2.2}$$

Therefore, selecting actions with the highest Q-value in state s will lead to higher cumulative rewards in the future.

Double Q-learning [28] is a reinforcement learning algorithm designed to mitigate overestimation bias in action value estimates, which can occur in traditional Q-learning algorithms. Overestimation bias can lead to sub-optimal or unstable learning in certain environments. The idea is to use two separate Q-value estimators, resulting in two sets of network weights $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. For each update, one set of weights is used to determine the greedy policy and the other to determine its value. For comparison, the temporal difference (TD) errors of single Q-learning and double Q-learning are:

$$\delta_t^Q = r_{t+1} + \gamma Q_{\boldsymbol{\theta}}(s_{t+1}, \arg\max_a Q_{\boldsymbol{\theta}}(s_{t+1}, a)) - Q_{\boldsymbol{\theta}}(s_t, a_t) \tag{2.3}$$

$$\delta_t^{\text{DoubleQ}} = r_{t+1} + \gamma Q_{\boldsymbol{\theta}}(s_{t+1}, \arg\max_a Q_{\boldsymbol{\theta}'}(s_{t+1}, a)) - Q_{\boldsymbol{\theta}}(s_t, a_t) \tag{2.4}$$

The first set of weights is updated using gradient descent. The second set of weights can be updated either by copying the first set of weights every N steps or via Polyak averaging $\boldsymbol{\theta'} \leftarrow \tau\boldsymbol{\theta} + (1-\tau)\boldsymbol{\theta'}$, with $\tau \in (0, 1)$. Using these independent estimators, an unbiased estimate of the action values can be obtained.

2.2.2 Soft-Actor Critic Algorithm

The Soft-Actor Critic algorithm (SAC) is an RL algorithm that combines the RL framework with the principle of maximum entropy and is designed for continuous action spaces. The policy seeks to maximise a modified version of the expected future reward which is defined as:

$$\max_{\pi} J(\pi) = \sum_{t=0} \mathbb{E}_{\pi}[r(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t))] \tag{2.5}$$

$J(\pi)$ maximises both the expected cumulative reward and an entropy term $\mathcal{H}(\pi(\cdot|s_t))$, to encourage exploration at the time of training and improve training speed. The parameter α is known as the temperature and it affects the weight of the entropy term. More precisely, SAC aims to learn three functions: the policy network with parameter θ, π_θ, a soft Q-value function parametised by w, Q_w, and a soft state value function parametrised by ψ, V_ψ. The experience gathered by the agent is stored in a replay buffer and, similar to DQN and DDPG, the Q network and the value network are trained using supervised learning with the data contained in a replay buffer. The targets for the network update are defined as:

$$\hat{Q}(s_t, a_t) = r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1}\sim\rho_{pi}(s)}\left[V_\psi(s_{t+1})\right] \tag{2.6}$$

$$\hat{V}(s_t) = \mathbb{E}_{a_t\sim\pi_\theta}\left[Q_w(s_t, a_t) - \alpha\log\pi_\theta(a_t|s_t)\right] \tag{2.7}$$

The policy is parametrised as a stochastic neural network

$$a_t = f_\theta(\epsilon_t, s_t) \tag{2.8}$$

where ϵ_t is an input noise vector, sampled from a Gaussian distribution. Then objective function for policy optimisation can be rewritten as:

$$J_\pi(\theta) = \mathbb{E}_{s_t,\epsilon_t}\left[\alpha\log(\pi_\theta(f_\theta(\epsilon_t, s_t)|s_t) - Q_\theta(s_t, f_\theta(\epsilon_t, s_t))\right] \tag{2.9}$$

2.2.3 Proximal Policy Optimisation Algorithm

Proximal Policy Optimisation (PPO) [29] is a model-free Deep RL algorithm designed for continuous actions spaces. In order to improve training stability, PPO imposes a constraint on the size of the policy update at each iteration, which results in smoother policies that are appealing when considering our problem from an ergonomics perspective.

The objective function measures the total advantage over the state visitation distribution and actions. In a standard off-policy algorithm it can be expressed as:

$$J(\theta) = \mathbb{E}_{a\sim\beta}[\frac{\pi_\theta(a|s)}{\beta(a|s)}A_{\theta_{old}}(s,a)]$$

where $\beta(a|s)$ is the sampling distribution.

Since PPO uses the old policy θ_{old} to generate data and we update the parameters θ, the objective function becomes:

$$J(\theta) = \mathbb{E}_{a\sim\pi_{\theta_{old}}}[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}A_{\theta_{old}}(s,a)] \tag{2.10}$$

PPO updates the policy parameters θ so as to maximise a slightly modified version of equation 2.10, which takes into account the constraint on the policy update, by the addition of a clipping parameter ϵ. Let $r(\theta) = \pi_\theta(a|s)/\pi_{\theta_{old}}(a|s)$, the modified objective function for PPO is:

$$J(\theta) = \mathbb{E}_{a\sim\pi_{\theta_{old}}}[\min(r(\theta)A_{\theta_{old}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)A_{\theta_{old}}(s,a))] \tag{2.11}$$

The objective of this modified objective function is to discourage policy updates that would cause big policy parameters variation even though they would lead to greater rewards.

2.3 Social Value Orientation

In Social Psychology, Social Value Orientation (SVO) [30], [31] is a value that describes how much a person values other people's welfare in relation to their own. Each individual can be modelled as an agent that selects actions so as to maximise their own utility function. We can model each individuals social preferences by expressing their own utility function as a combination of two terms, the ego agent's selfish utility U_{self} and a term that captures other agents' utility U_{other} (see Fig 2.3):

$$U_{total} = \cos(\varphi)U_{self} + \sin(\varphi)U_{other} \tag{2.12}$$

where φ is the SVO value. It is an angle, whose value affects the weights of the two utility terms, and therefore the balance between selfish and altruistic rewards. We can characterise the personality of each individual with the SVO value. For example, an SVO value of 90° corresponds to fully altruistic behaviour, whereas an SVO value of 0° corresponds to an individualistic agent. In our work, we focused on SVO values between 0° and 90°, as we want the AV to exhibit pro-social behaviour and yield to the pedestrian if necessary to avoid dangerous situations.

SVO has been previously used to design controllers in a game-theoretic setting [32], but this demands long complex computations to solve for a Nash equilibrium

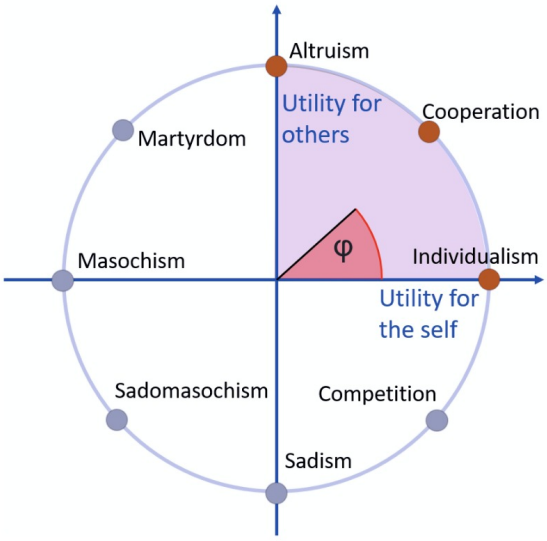


Figure 2.3: Social Value Orientation ring. The SVO value φ affects the behaviour of the ego-vehicle.

points. In our work, we try to mitigate the computational cost of the optimisation problem by using SVO in the RL framework, thereby moving the computational cost from execution time to training time, in a learning-based fashion. We integrate the SVO concept directly in the MDP model formulation by constructing a reward function that is composed of two terms, one that models the AV's own objective U_{self} and one that models the pedestrian's objective U_{other}.

2.4 Social Force Pedestrian Models

Social Force Models [26] are a class of mathematical models used to simulate the movement and behaviour of pedestrians in crowded environments. These models are particularly relevant for understanding the dynamics of pedestrian flow in situations like urban planning, building design, evacuation planning, and crowd management. The main idea behind Social Force Models is to represent pedestrian behaviour as a result of forces acting on individuals within a crowd. Although they have initially been designed for crowd studies, Social Force Models have also been extended to include vehicle influences on pedestrian motion [24].

In a Social Force Based model , pedestrians are regarded as point mass particles, with their motion governed by Newton equations of motion:

$$\frac{d^2\vec{\mathbf{r}}}{dt^2} = \frac{d\vec{\mathbf{v}}}{dt} = \frac{\vec{\mathbf{F}}_{total}}{m} \tag{2.13}$$

where $\vec{\mathbf{r}}$ and $\vec{\mathbf{v}}$ represent the pedestrian position and velocity respectively, and m represents the pedestrian's mass. The total force $\vec{\mathbf{F}}_{total}$ influences the pedestrian

acceleration and can be decomposed further in three terms:

$$\vec{\mathbf{F}}_{total} = \vec{\mathbf{F}}_{nav} + \vec{\mathbf{F}}_{veh} + \vec{\mathbf{F}}_{soc} \tag{2.14}$$

The three terms have different effects on the pedestrian motion and shape the pedestrian's trajectory, making them reach their goal position while avoiding obstacles at the same time. The term $\vec{\mathbf{F}}_{nav}$ has the overall effect of pulling pedestrians towards their goal position. The term $\vec{\mathbf{F}}_{veh}$ is used to shape the effect of the vehicle on the pedestrian motion, whereas the term $\vec{\mathbf{F}}_{soc}$ is the so called social force. The social force term models how pedestrians interact with each other but since we are mainly concerned on AV decision-making in the presence of a single pedestrian, we will neglect this term in the further discussions within this paper.

2.5 Graph Neural Networks

Graph Neural Networks (GNNs) are a class of Neural Networks (NNs) models that have been developed to process graph-structured data. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a data structure that consists of a set of nodes (vertices) $\mathcal{V}$ and a set of edges $\mathcal{E}$ that connect pairs of nodes. Graphs are used to represent and model various types of relationships and connections between objects or entities.

Nodes are identified by a unique index $i \in \mathcal{V}$ and directed edges as ordered pairs of nodes (i, j). Nodes and edges can be associated to feature vectors $\mathbf{x}_i$ and $\mathbf{x}_{i,j}$. The feature vectors contain some characteristics of the nodes or of the edges. As an example, a graph could be used to represent the road structure connecting towns on a map. In this case, nodes can be associated to cities and edges to the roads connecting them. Node features would include city related information, such as population and area, whereas edge feature would represent road information, such as the length of the road. A GNN processes a graph with its associated feature vectors and make predictions. This kind of predictions can be performed at node-level, edge-level, or graph-level. Graph-level predictions concerns global-properties of the graph, whereas node-level and edge-level predictions are predictions for each node and edge in the graph respectively. Each node has an associated set of neighbours $\mathcal{N}(i)$, which consists of all the nodes j such that an edge from j to i exists, $(\mathcal{N}(i) = \{j|(j, i) \in \mathcal{E}\})$.

A key feature regarding GNNs (see Fig.2.4) is the message-passing operation. It is a similar operation to the convolution used for image processing in Convolutional Neural Networks and is used to combine information from different nodes to improve the predictions. Message passing operations include GCNConv (Graph Convolutional Networks) [33], GAT (Graph Attention Networks) [34] and GraphSAGE [34]. Common to all of them is that they update features of each node and edges by combining their features with those of neighbouring nodes. We denote hidden representations in the neural network for nodes and edges as $\mathbf{h}_i$ and $\mathbf{h}_{i,j}$, and updated representations as $\mathbf{h}'_i$ and $\mathbf{h}'_{i,j}$. We can write the operations as:

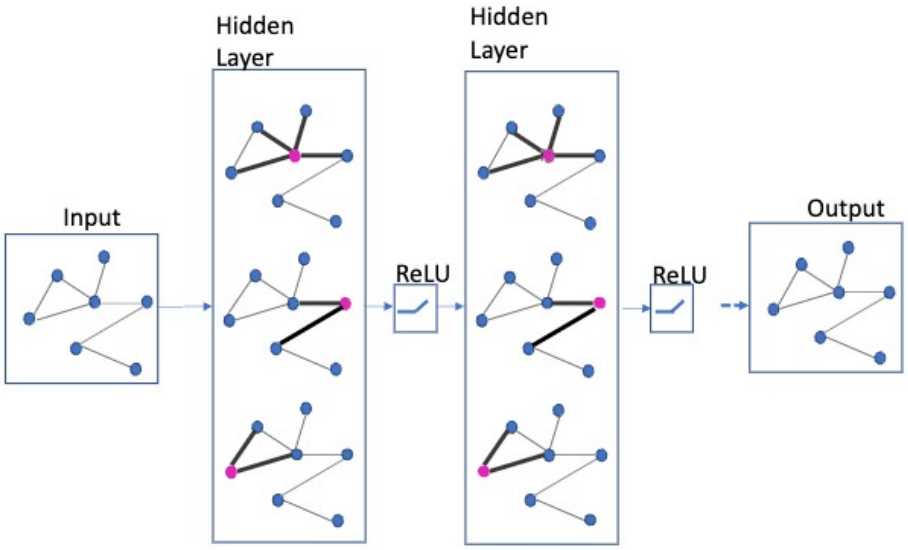


Figure 2.4: Example of a graph neural network.

$$\mathbf{h}_i' = \phi\left(\mathbf{h}_i, \bigoplus_{j \in \mathcal{N}_i} \psi\left(\mathbf{h_i}, \mathbf{h_j}, \mathbf{h_{i,j}}\right)\right) \tag{2.15}$$

where ϕ and ψ are differentiable functions (e.g. artificial neural networks whose parameters need to be learned) and $\bigoplus$ is a permutation-invariant aggregation operation over neighbouring nodes that can accept any number of inputs (e.g. element-wise sum, min, or max). A similar general expression can also be used to update edge feature representations but will be less relevant to this thesis, therefore will be omitted.

GCNs can process dynamically-changing graphs and can capture interrelationships between nodes, which makes them ideal candidates to study in the field of interaction-aware decision-making for AVs. They will serve as the basis for multi-agent scenarios studies in Chapter 5.

2.6 Level-k Game Theory

Game theory models have been introduced in 3.3.3 of Chapter 3. Level-k game theory, also known as cognitive hierarchy theory, is a concept in game theory that seeks to model the decision-making process of individuals in strategic interactions. At the lowest tier of this hierarchy is what it is called level-0 reasoning. A level-0 agent can be described as nonstrategic or naive since their decisions do not take into consideration other agents' potential actions; instead, they rely on pre-determined actions. Moving up one level, we encounter strategic level-1 agents. These individuals determine their actions by assuming that other agents are operating at level-0 reasoning. Consequently, the choices made by level-1 agents are optimal responses to the actions

of level-0 agents. In a similar fashion, level-2 agents perceive other agents as level-1 reasoners and make their decisions accordingly. This sequential pattern continues for higher reasoning levels. Notably, in certain experiments, researchers have observed that humans typically exhibit reasoning levels up to level-3, although this can vary depending on the specific type of game being played.

Level-k game theory has the advantage of computing approximations of Nash Equilibrium by reducing the reasoning depth. It can help describe and explain deviations from Nash equilibrium, especially in cases where players have bounded rationality or varying levels of strategic sophistication. Level-k reasoning provides a behavioural perspective on how individuals approach strategic interactions, which can shed light on real-world decision-making that does not always align with the strict assumptions of Nash equilibrium.

Chapter 3

Literature Review

A large variety of methods ranging from Game Theoretic Models to Machine Learning approaches has been applied to the problem of Autonomous Driving. In this Chapter, I provide an overview of such methods, with a focus on AV-Pedestrian Interaction.

This review consists of three main sections which cover different areas in interaction-aware autonomous driving. We introduce terminology used in interaction-aware autonomous driving in Section 3.1. Section 3.2 will cover human factors studies on what affects human decision making while driving, as well as pedestrian behavioural studies. Section 3.3 gives a broad overview and classification of existing techniques that are used in interaction modelling. Sections 3.3.1 and 3.3.2 cover state-of-the-art techniques used for motion-planning and decision-making in interactive scenarios.

3.1 Terminology

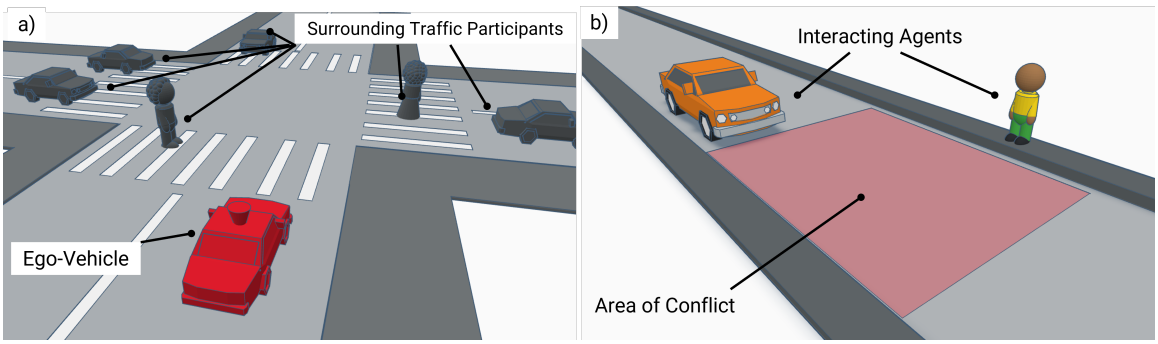


Figure 3.1: a) the ego-vehicle is controlled by the autonomous system, whereas surrounding traffic participants act on their own will. b) two agents interacting with each other determine an area of conflict.

Before we discuss the recent advances in interaction-aware motion-planning and decision-making, we will first define some of the terminology used in this field. In the

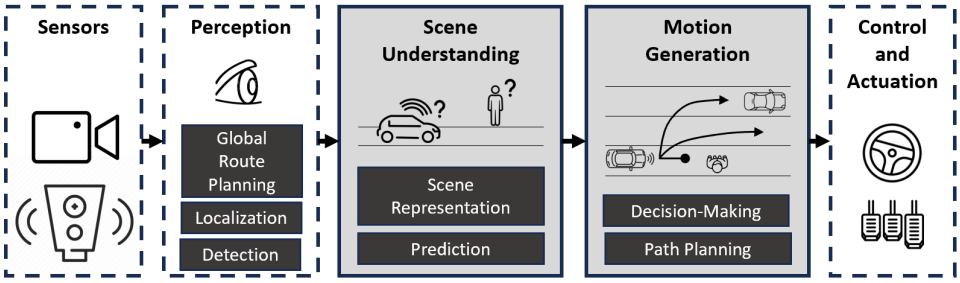


Figure 3.2: Architecture of AV systems. Solid line boxes identify modules that are closely related to interaction-aware models.

field of autonomous driving, the term *ego-vehicle* refers to the specific vehicle whose behaviour is to be controlled and studied. All other vehicles, cyclists, pedestrians, etc. that occupy a region of space around the ego-vehicle are treated as interactive obstacles and are referred to as *surrounding traffic participants* or as *other road users*, see Fig. 3.1a. Interaction-aware autonomous driving is a field of research that focuses on developing AVs that can safely and efficiently interact with other road users, such as vehicles, pedestrians, and cyclists. Traditional autonomous driving approaches often treat *surrounding traffic participants* as dynamic obstacles. However, this is not a realistic approach, as they are constantly changing their behaviour to adapt to the current situation. Multiple surrounding traffic participants can give rise to *space-sharing conflicts* amongst themselves or with the ego-vehicle: a situation from which it can be reasonably inferred that two or more road users are intending to occupy the same region of space at the same time in the near future, see Fig. 3.1b. The agents involved in the conflict are said to display an *interactive behaviour*, which implies that their behaviour would have been different if the space-sharing conflict had not occurred [35]. Since space sharing conflicts happen all the time when driving, it is crucial that the algorithms developed for AVs be aware of the dynamics of the interactions between agents. Such algorithms are said to be *interaction-aware* and are often the focus of recent autonomous driving research [36].

There are a number of challenges that need to be addressed in order to develop interaction-aware autonomous driving systems [37]. One challenge is the need to accurately predict the behaviour of other road users. This is a difficult task, as the behaviour of other road can be affected by a variety of factors. Another challenge is the need to develop algorithms that can safely and efficiently interact with other road users and produce an AV behaviour that compels to human-like standards. Fig. 3.2 shows the main parts that make up an AV system. Raw data from sensors is processed by a Perception Module, which detects the surrounding environment and performs localisation, which allows generating a global-route plan for the ego-vehicle to reach its target destination. The scene can be further interpreted and predictions regarding surrounding traffic participants can be performed. Interaction-aware models play a major role in prediction tasks, as agents affect each other's trajectory and decisions.

Decision-making and path-planning are two of the most important tasks in autonomous driving. They are responsible for determining how the vehicle will move through its environment. Decision-making is the process of choosing an action from a set of possible options. For example, the vehicle may need to decide whether to change lanes, slow down, or stop. Path-planning is the process of generating a safe and feasible trajectory for the vehicle to follow. Decision-making and path-planning are closely related. The decision-making process typically outputs a high-level plan, such as "change lanes to the left." The path-planning process then takes this plan and generates a detailed trajectory that the vehicle can follow. Both tasks must take into account the vehicle's current position, the vehicle's capabilities and the surrounding traffic, which is why interaction-aware models are highly relevant to these two tasks. From a control system perspective, the dynamics of the vehicle are represented by its states, i.e. position and orientation, and their time derivatives. The state of the environment is determined by the states of all dynamics and static entities. The strictly physical state-space can also be augmented with additional latent-space variables that capture, for example, the intentions [38] or the behavioural preferences of surrounding users [32], which are part of the Scene Understanding system.

3.2 Human Behaviour Studies

This section synthesizes empirical and modelling research findings on HRU behaviour, including that of human drivers and pedestrians, interacting with AVs or conventional vehicles, especially from a communication perspective. We focus on research involving road interactions relevant for Chapter 4 and for the Virtual Reality framework introduced in Chapter 6, with the aim of discovering insights that may facilitate the development of *interaction-aware AVs*. Studies that look at macro-traffic conditions, such as the influences of route choice, weather, or regulation, beyond this thesis scope. The studies in this Section are of particular interest for RQ1, RQ2, and RQ4 introduced in Chapter 1.

Since road traffic is unlikely to become fully automated in the near future, AVs will inevitably operate in mixed environments with human road users (HRUs), including human drivers and pedestrians. This has raised concerns that AVs' inability to understand and interact smoothly with HRUs' may cause traffic dilemmas and safety issues [39]. However, the safe and socially acceptable deployment of AVs into these complex interactive environments is currently hampered by a lack of innovative theories about how human road users interact [40]. The theories to be developed are not limited to predicting and modelling HRU behaviour but also exploring behaviour patterns and underlying psychological mechanisms of HRU behaviour. Integrating AVs into road traffic as seamlessly as humans would require more advanced behaviour models.

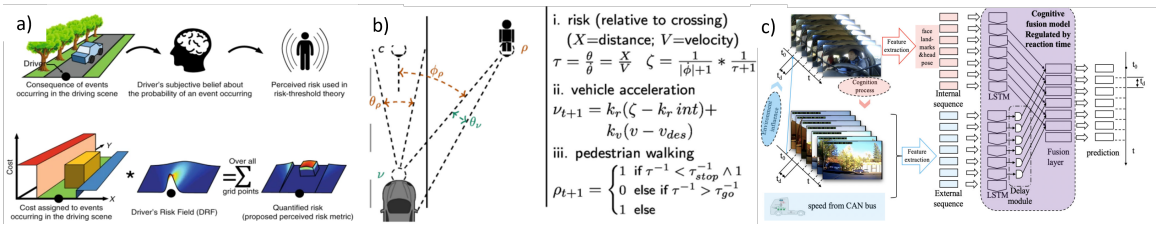


Figure 3.3: Illustration of Driver Behaviour Models. a) driver risk field from [43], b) joint theory-based model in [44], c) data-driven model in [45]

3.2.1 Driver Behaviour Studies

Driver behaviour models are used to predict and understand how drivers will behave in different driving scenarios. These models can be used to improve the safety and efficiency of transportation systems and aid in the process of designing AVs. Many different factors can affect driving behaviour, including individual characteristics (age, gender, personality, experience), environmental factors, i.e. road and weather conditions, and social factors, which include the driver's interactions with other road users [41]. A comprehensive overview of DBM in vehicle-vehicle interactions can be found in [42]. Here, we will focus on DBMs that are relevant to vehicle-pedestrian interactions.

The most common driver behavioural models include:

- *Driver's Risk Field Models*: (Fig. 3.3a) This model predicts how drivers will perceive risk in different driving situations. The DRF model is based on the idea that drivers make decisions based on their perception of risk. The results of [43] suggest that driving behaviour is governed by a cost function that takes into account the effects of noise in human perception and actions. This is similar to how motor-control tasks are governed by cost functions. Risk perception onboard of AVs has also been analysed in [46] in a driving simulator scenarios.

- *Theory Based*: (Fig. 3.3b) perceptual and cognitive models. Models based on perceptual information describe driver's behaviour based on perceptual cues, e.g. distance, vehicle speed, acceleration, expansion angle, reaction times, etc [44], [47]. Cognitive models outline the internal state flow and motive that regulates driver's behaviour as a psychological human being [48], [49].

- *Data Driven Models*: (Fig. 3.3c) this set of methods relies on analysing naturalistic driving data with machine learning to analyse driver behaviour. Data-Driven Models can learn a generative or discriminative [45], [50] models of human behaviour to make predictions about driver's future decisions or preferred driving style. Model validation can be done by comparing predictions with real data and by human-in-the-loop simulations.

Existing research highlights based on naturalistic driving data analyses how drivers behave in the presence of pedestrians. In [51], the authors found that drivers tend to

maintain smaller minimum lateral clearance and lower overtaking speed when overtaking pedestrians who are walking in the opposite direction, on the lane edge, or when oncoming traffic is present. Minimum lateral clearance and time-to-collision were only weakly correlated with overtaking speed. The results in [52] show that the vehicle deceleration behaviour is relative to initial TTC, subjective judgment of pedestrian crossing intention, vehicle speed, pedestrian position and crossing direction.

There is less attention paid to multi-agent settings where multiple vehicles and pedestrians interact with each other. In [53], the authors develop a Multi-agent adversarial Inverse Reinforcement Learning (IRL) framework based on data collected at a road intersection to simulate driver and pedestrian behaviour at intersections.

Overall, DBMs are a promising area of research with the potential to significantly improve the safety and efficiency of transportation systems. However, there is still much work to be done in developing and validating these models. Future research should focus on developing more comprehensive models that take into account a wider range of factors, such as the driver's internal state, the environment, and the interactions with other road users.

3.2.2 Pedestrians Behaviour Studies

Since pedestrians are considered the most vulnerable road users, lacking protective equipment and moving more slowly than other road users [54], investigating pedestrian behaviour is clearly relevant to the safety and acceptance of AVs interacting with pedestrians. Pedestrian behaviour has been the subject of extensive research for decades [55]. The emergence of AVs has recently prompted many new research questions about pedestrian behaviour. Given the large body of work in this area and our aims, this Section examines major studies rather than providing an exhaustive survey. We review pedestrian behaviour studies regarding interactions with vehicles from three perspectives: communications, theories and models of crossing behaviour, and AV-involved applications. We aim to identify and summarise their value for developing interaction-aware AVs.

Communications

Pedestrian behavior in road-crossing scenarios is significantly influenced by the kinematics and signaling information of autonomous vehicles (AVs), particularly in the absence of a human driver [15], [35], [56]. Research suggests that understanding critical motion cues or signals, such as speed, distance, and braking, is essential, as these factors impact pedestrian decision-making and safety [57].

Implicit Communication Signals like vehicle speed, distance, and braking maneuvers serve as cues that pedestrians use to interpret the intentions of vehicles [35]. Studies indicate that pedestrians feel safer crossing when vehicles are farther away or moving slower. However, they tend to rely more on distance rather than time to

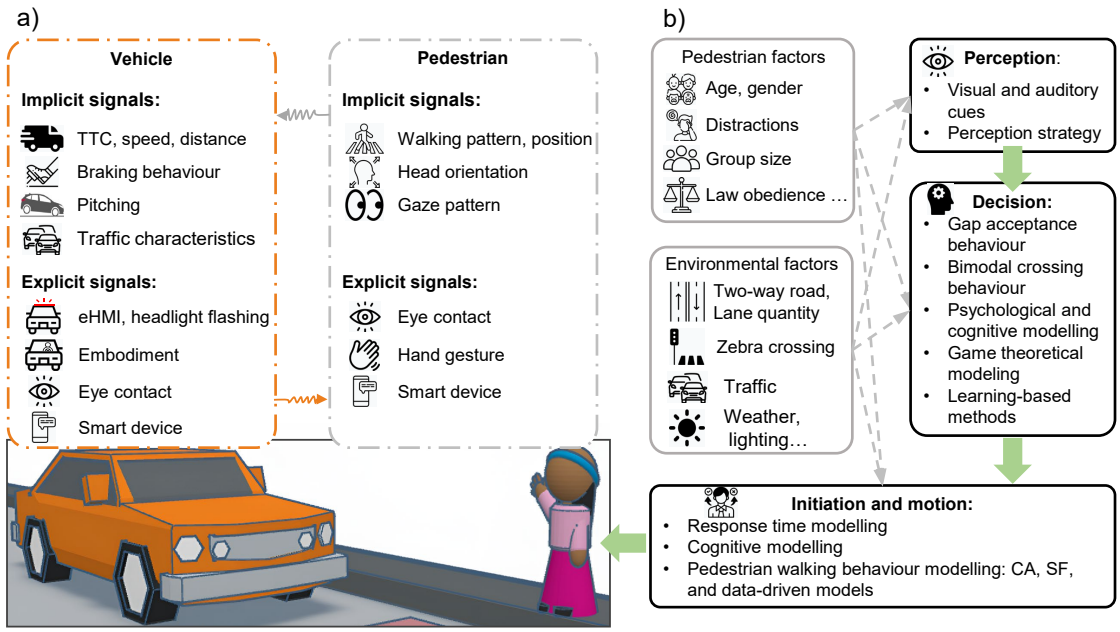


Figure 3.4: a) Communication between pedestrians and automated vehicles. b) Theories and models for pedestrian crossing perception, decision, initiation, and motion.

collision (TTC) when making crossing decisions, suggesting that multiple kinematic factors are considered simultaneously [58]. Pedestrian behavior also responds to vehicle braking patterns; for instance, early and gentle braking increases pedestrian comfort, while harsh braking leads to avoidance [59], [60].

Pedestrians gather additional implicit information from traffic characteristics, such as traffic volume, which affects their crossing behavior. High traffic volumes may force pedestrians to take smaller gaps, increasing risk-taking behavior over time [61], [62]. Pedestrian movements, such as stepping onto the road or making eye contact, can also signal intentions to vehicles [63].

Explicit Communication Signals involve deliberate actions, like using external human-machine interfaces (eHMIs) to convey messages to pedestrians. For example, AVs might use light signals or text to indicate their intentions, such as yielding [59], [64]. However, the effectiveness of eHMIs varies depending on pedestrian familiarity and environmental factors like weather conditions [65]. Some studies suggest that pedestrians may rely more on implicit cues than on eHMIs due to reliability concerns [66].

From the perspective of pedestrians, explicit signals such as eye contact or hand gestures can communicate their intent to drivers or AVs, although AVs might lack the capacity to respond like human drivers [67]. Solutions, such as a visual embodiment of a driver or enhanced wireless communication via smart devices, could improve interactions between pedestrians and AVs [68], [69].

Theories and Models of Crossing Behavior: Pedestrian crossing behavior

involves multiple cognitive processes, typically framed within the situation awareness model, which includes perception, decision-making, and action initiation. This model explains how pedestrians assess vehicle features and environmental cues, integrate them with prior knowledge, and make decisions accordingly [70]. Further sections will explore these processes in detail (Fig. 3.4).

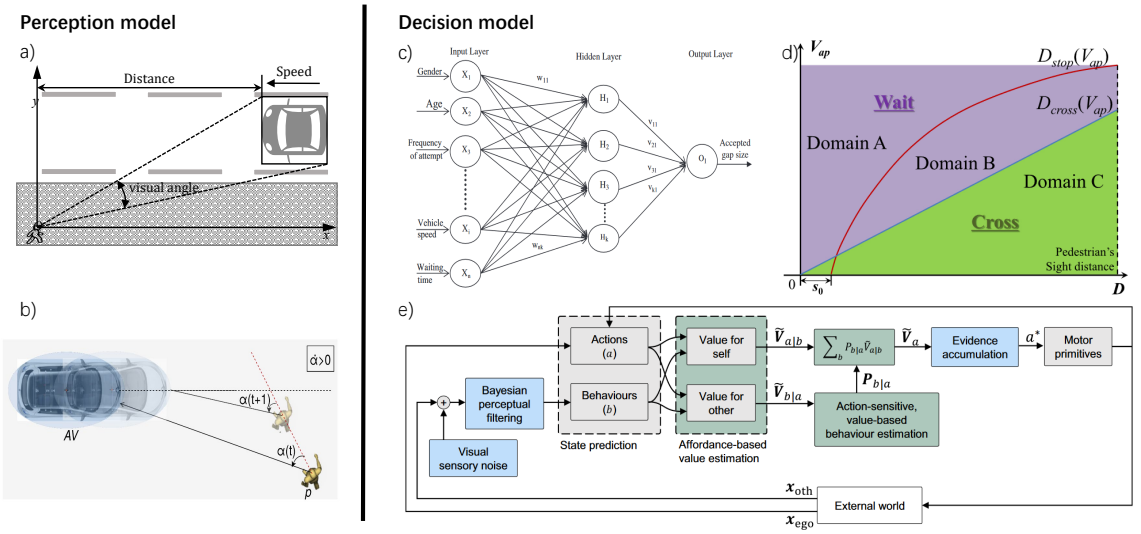


Figure 3.5: Perception and decision models for pedestrians. a) Visual cues, including $\theta, \dot{\theta}, \tau, \dot{\tau}$ [58], [71]. b) Bearing angle [72]. c) Artificial neural networks [73]. d) Speed-distance model [74]. e) Large computational psychological model [40].

Perception Visual perception is key to how pedestrians detect approaching vehicles. As an object nears, its image on the retina enlarges, forming the basis for human collision perception, known as the visual looming phenomenon [75], [76]. The rate of change of the visual angle, $\dot{\theta}$, helps pedestrians judge a vehicle's approach, but it lacks temporal information on when the vehicle will arrive [77]. The ratio τ (visual angle to its rate of change) and its derivative $\dot{\tau}$ provide time-to-collision (TTC) cues critical for assessing if a vehicle can stop in time [25], [78]. Pedestrians also use simpler cues like θ and $\dot{\theta}$ when vehicles are farther away and more complex cues like τ for closer, imminent collisions [77]. Although visual information is primary, auditory cues can affect perception; for instance, quiet vehicles lead to overestimated TTCs [79]. Factors such as age, distractions, and sensory limitations also influence perception, with older adults and children at higher risk due to limited perceptual abilities [80], [81].

Decision Pedestrian crossing decisions depend on vehicle interactions, traffic conditions, and individual differences. At uncontrolled crossings, pedestrians decide based on the gaps between vehicles, often modeled as "gap acceptance" behavior using critical gap or binary logit models [82], [83]. In scenarios with yielding vehicles, decisions

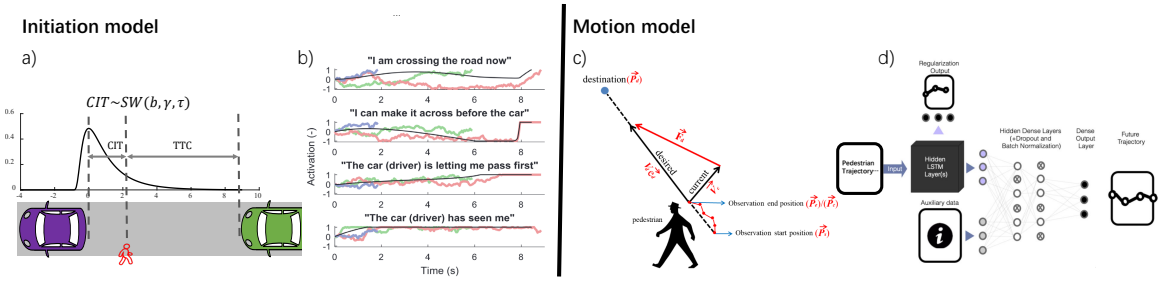


Figure 3.6: Initiation and motion models for pedestrians. a) Response time model [62]. b) Evidence accumulation model [35]. c) Social force model [102]. d) LSTM-ANN [101].

follow a bimodal pattern, where pedestrians prefer crossing when gaps are large or when vehicles are stopping [84], [85]. Models such as reinforcement learning, evidence accumulation, and game theory have been used to describe these decisions based on visual cues, cognitive processes, and dynamic negotiations with vehicles [40], [58], [86].

Pedestrian decisions are also influenced by environmental complexity, such as multi-lane crossings, intersections, and traffic flow, where waiting time and risk acceptance can vary significantly [61], [62]. Individual differences like age, distractions, and group behavior also impact decision-making, with older pedestrians often choosing smaller gaps and distractions reducing decision accuracy [73], [87], [88].

Initiation and motion The duration before pedestrians start to cross is called the crossing initiation time (CIT), reflecting the time-dynamic nature of decisions [103]. Depending on the formation of traffic gaps, CIT has different definitions. When pedestrians cross a gap between two approaching vehicles, CIT is the duration between the moment when the rear end of the previous vehicle passes the pedestrian's position and when the pedestrian begins to move [58], [104]. Alternatively, if the traffic gap is between pedestrians and an approaching vehicle, CIT is the duration between when the vehicle first appears in the lane and when pedestrians begin to cross [85]. According to drift diffusion theory [105], CIT is a variable influenced by the accumulation process of noisy evidence in the cognitive system and may reflect the efficiency of pedestrian cognitive and locomotor systems. It was found that vehicle kinematics, age, gender, and distractions affected CITs. Pedestrians tended to initiate slower at higher vehicle speed conditions [58]. Female pedestrians initiated quicker than males [106]. Elderly pedestrians initiated sooner than young pedestrians [107]. Distraction's impact on CITs differently, depending on its components [108].

When pedestrians encounter vehicles that do not yield, the risk of collision increases as the distance between the vehicle and the pedestrian decreases. Therefore, taking too much time to decide increases the chances of missing crossing opportunities. Pedestrians in such situations usually make decisions quickly by taking 'snapshots'

Table 3.1: Pedestrian models and theories

Research	Scenario	Cognitive process	Models	Theories	Considered factors	Action
[25]	1,3,4	Perception	τ, $\dot{\tau}$	Visual perception	Vehicle kinematics, eHMI, eye contact	Continuous
[89]	1,3,4	Perception	τ, $\dot{\tau}$	Visual perception	Vehicle kinematics	Continuous
[58], [62]	1,3	Perception	$\dot{\theta}$	Visual perception	Vehicle kinematics	Discrete
[44]	1,4	Perception	τ, bearing angle	Visual perception	Vehicle kinematics	Continuous
[85]	1,3,4	Perception	Generalised TTC	Visual perception	Vehicle kinematics, eHMI	Continuous
[72]	1,4	Perception	TTC, bearing angle	Visual perception	Vehicle kinematics	Continuous
[90]	1,3	Perception	Perceived distance	Visual perception	Vehicle kinematics	Continuous
[91]	1,3	Decision	Critical gap	GA behaviour	Vehicle kinematics	Discrete
[92] [93]	1,3	Decision	Critical gap	GA behaviour	Vehicle kinematics, pedestrian speed, road length	Discrete
[62] [61]	1,3,5	Decision	LR	GA behaviour	Vehicle kinematics, waiting time, group size, pedestrian position	Discrete
[73]	1,3	Decision	ANN	GA behaviour Machine learning	Vehicle kinematics, Waiting time, pedestrian age, cellphone usage, rolling gap, group size	Discrete
[94]	1,2,3, 4,5	Decision	Critical gap	GA behaviour	Vehicle kinematics, pedestrian age, traits, law obedience, rolling gap, crossing	Continuous
[90]	1,3	Decision	RL, Bayesian filter	GA behaviour, learning-based	Vehicle kinematics	Continuous
[74]	2,3,4	Decision	Speed-distance	BC behaviour	Vehicle kinematics	Discrete
[71]	1,3,4	Decision	Hybrid perception, LR	Visual perception, BC behaviour	Vehicle kinematics	Continuous
[25], [85] [89]	1,3,4	Decision	EA	Drift diffusion	Vehicle kinematics, eHMI, eye contact	Continuous
[40]	1,3,4	Decision	EA, Bayesian filter	Drift diffusion, Game theory, Theory of Mind, Noisy visual perception	Vehicle kinematics	Continuous
[95]	1,3,4	Decision	DA game	Game theory	Vehicle kinematics	Continuous
[96]	1,4	Decision	SC game	Game theory	Vehicle kinematics	Continuous
[72]	1,4	Decision	Critical gap	GA behaviour, visual perception	Vehicle kinematics, interaction angle	Continuous
[25], [40], [85] [89]	1,3,4	Initiation	EA	Drift diffusion	Vehicle kinematics	n/a
[62] [71]	1,3,4	Initiation	SW distribution	Response time	Vehicle kinematics	n/a
[90]	1,3	Initiation	RL	Learning-based	Vehicle kinematics	n/a
[72] [97] [24]	1,3,4	Motion	SF	Walking behaviour	Vehicle kinematics, road structure, crossing	Continuous
[98]	2	Motion	ANN	Learning-based	Pedestrian kinematics	Continuous
[99] [100]	1,2,3,4	Motion	CA	Walking behaviour	Road structure, vehicle kinematics	Discrete
[53]	2,3,4	Motion	Adversarial IRL	Learning-based	Vehicle and pedestrian kinematics	Continuous
[101]	2,4	Motion	LSTM	Learning-based	Vehicle and pedestrian kinematics	Continuous

1. Uncontrolled crossings. 2. Controlled crossings. 3. With non-yielding vehicles. 4. With yielding vehicles. 5. With traffic flow.

of approaching vehicles [84], [104], and the distribution of CIT along the time axis is typically concentrated and right-skewed [89]. To model CITs in such scenarios, response time models can be used, which are typically closed-form probability density functions with right skews, such as Ex-Gaussian and Shifted Wald (SW) distributions [109]. [110] modelled CITs as variables following SW distribution (Fig. 3.6a). Moreover, EA models describe CIT distribution through the accumulation process of noisy evidence [25], [85], [89], [111] (Fig. 3.6b). Furthermore, [90] applied an RL model to learn the crossing initiation pattern of pedestrians.

Regarding CITs in vehicle-yielding scenarios, CITs present a bimodal distribution. [85] indicated that CITs in these scenarios could be categorised into early and late groups. For the early group, [112] found that the distribution of CITs was similar to that of CITs in non-yielding scenarios, as pedestrians might use the same decision-making strategy in this phase of vehicle-yielding scenarios as in non-yielding scenarios. For the late group, The distribution of CITs has a complex shape that is difficult to describe with the common response time distribution [85]. To solve this problem, [85], [89] proposed EA models with time vary evidence. The type and intensity of evidence varied over time, which enabled the EA model to generate CIT distributions with complex shapes. Furthermore, [71] modelled CITs in vehicle-yielding scenarios by assuming that CITs obeyed the joint distribution of response times distribution for each pedestrian.

After pedestrians initiate their crossing decisions, they need to walk to the opposite side of the road. Walking is a key part of crossing behaviour and is influenced by many factors. Approaching vehicles prompted pedestrians to change their walking trajectories to stay away from vehicles [97]. At multi-lane crossings, pedestrians tended to walk to and wait at lane lines and accepted the traffic gap in each lane successively [113]. It was found that pedestrian walking speeds at crossings were faster than normal walking speeds in other scenarios [114]. Although previous studies found no significant effect of gender on walking speeds, teen and elderly pedestrians walked slower than young and middle adults [114], [115]. Distractions are another important influential factor that could reduce pedestrian walking speeds [108].

The walking behaviour can be simulated using microscopic pedestrian motion models, which include three main types: Cellular Automata (CA) models, Social Force (SF) models, and learning-based approaches. CA models consist of finite state cells on a uniform grid [116]. The state of each cell depends on a set of rules that determine its new state based on the current state of the cell and its neighbours. CA models are discrete in space, time, and state, making them ideal for simulating complex dynamic systems such as pedestrian-vehicle interactions. [99], [100] applied CA models to simulate pedestrian crossing behaviour at multi-lane intersections. The SF models, based on Newton's second law, assume that pedestrians are driven by a desired force from their destination, repulsive and attractive forces from vehicles, other pedestrians, or traffic signals, and are restricted by boundaries such as the edge of crossings [26], [97], [117]. The SF models are commonly used to simulate

Table 3.2: Applications of pedestrian theories and models in AV contexts

Research	**Purpose**	**Applied theory**	**Applied model**
[120]	Pedestrian trajectory prediction	Learning-based	GCN
[121]	Pedestrian trajectory prediction	Learning-based	SVM, LSTM, Dense NN
[101]	Pedestrian trajectory prediction	Learning-based	LSTM
[102]	Pedestrian trajectory prediction	GA behaviour, walking behaviour	LR,SF
[72]	Pedestrian behaviour modelling	Visual perception, GA behaviour, walking behaviour	Critical gap SF bearing angle
[122]	Pedestrian behaviour modelling	GA behaviour, walking behaviour	Critical gap, CA
[74]	Pedestrian behaviour modelling	BC behaviour	Speed-distance
[96]	Pedestrian behaviour modelling	Game theory	SC game
[44]	Pedestrian behaviour modelling	Visual perception, GA behaviour	Critical gap τ bearing angle
[123]	Pedestrian behaviour modelling	GA behaviour, Game theory, walking behaviour	LR, critical gap, Stackelberg game, SF

large-scale pedestrian flows in evacuation scenarios [26] (Fig. 3.6c). Thanks to its ability to describe the interactions between agents through 'forces', SF models have been used to simulate pedestrian-vehicle interactions. [97] characterised pedestrian crossing behaviour at a signalised intersection using an SF model. [24] used an SF model to simulate the crossing behaviour of pedestrian crowds in complex interaction scenarios involving low-speed vehicles.

The pedestrian motion models discussed above are either discrete or continuous dynamical models based on interpretable empirical observations. In contrast to these *white box* models, there are *black box* models based on learning-based approaches, which learn pedestrian walking behaviour from naturalistic datasets or in pre-defined environments. For example, [98] employed ANNs to learn pedestrian walking behaviour by incorporating the relative spatial and motion relationships between pedestrians and other objects extracted from videos. [118] used the outputs of an SF model as inputs to ANNs to simulate multiple pedestrian walking behaviour. [101] proposed a Long Short-Term Memory Network (LSTM) pedestrian trajectory prediction model(Fig. 3.6d). Additionally, RL and IRL models were also applied to model pedestrian walking behaviour. [119] applied an RL model to learn multiple pedestrians walking behaviour in an SF environment. [53] developed an IRL model to learn pedestrian walking behaviour from video datasets.

AV-involved applications

In recent years, there has been an increasing interest in the interaction between AVs and pedestrians, leading to numerous studies that apply pedestrian crossing behaviour theories and models to improve or evaluate the performance of AVs in interactions. The most common approaches involve learning-based methods, which learn pedestrian intention and trajectory from real-world datasets to aid AVs' decision-making. For instance, [120] proposed a Graph Convolutional Neural Network-based pedestrian trajectory prediction model for generic AV Use Cases. This model used past pedestrian trajectories as the inputs to predict deterministic and probabilistic future trajectories. Other similar models aimed to improve prediction accuracy by considering the social context of interactions. For example, [101] proposed an LSTM pedestrian trajectory prediction model, which considered past trajectories, pedestrian head orientations, and distance to the approaching vehicle as the inputs to the model, as pedestrian head orientations and distance to approaching vehicles may be correlated with pedestrian awareness and perceived collision risks [15]. In addition to pedestrian trajectories prediction for AVs, many studies aimed to anticipate pedestrian crossing intentions. [121] applied SVM, LSTM, and ANN to predict pedestrian crossing intentions separately and found that the ANN outperformed SVM.

Learning-based approaches could accurately predict pedestrian trajectories and intentions. However, these approaches require a significant amount of data to achieve robust performance, which limits their scalability to interaction cases with insufficient data. Moreover, the *black box* nature of these models makes interpreting the generated trajectories and intents challenging, which poses a problem for AV decision-making modelling [72]. To address these issues, expert models have been developed. For example, the SF model was modified to predict pedestrian trajectories for AVs by incorporating more interaction details, such as TTC and the interaction angle between vehicles and pedestrians [72], [102]. Moreover, the SF and CA models were used to represent pedestrian crossing behaviour and embedded in the AV decision module to guide decisions of AV in interactions with pedestrians [122], [123].

Furthermore, crossing decision models have also been applied in AV research. [122] employed crossing critical gap models to characterise pedestrian crossing decisions in their AV decision module. [74] applied their speed-distance model to design defensive and competitive interaction behaviour for AVs. To enhance the dynamic and interactive nature of crossing decisions, game theoretical models were used to model crossing decisions when negotiating the right of way with AVs [96], [123]. Researchers also attempted to use pedestrian perception theories or models to design AV decision-making strategies. For instance, [44] simulated AV-pedestrian coupling behaviour using visual cues, τ and bearing angle, based on control theory. [72] modelled the right of way of AVs and pedestrians using bearing angle.

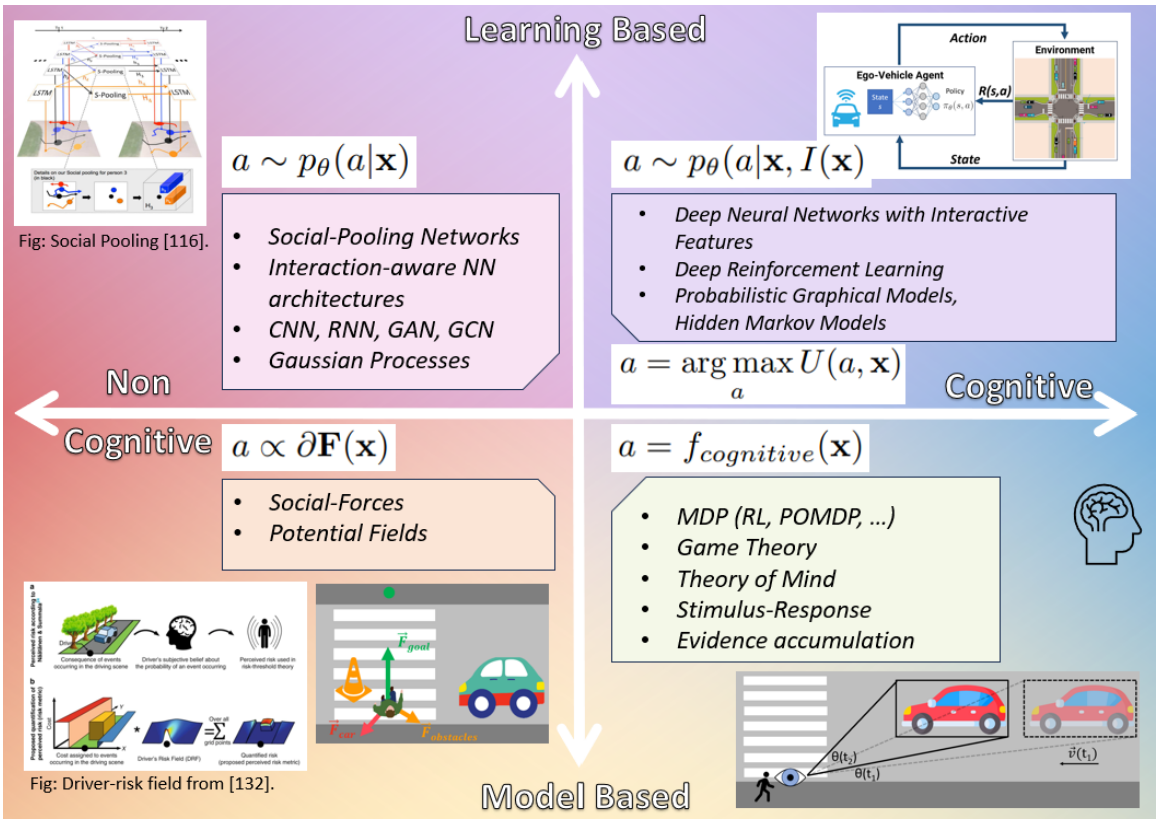


Figure 3.7: A map of state-of-the-art techniques in interaction-aware autonomous driving.

3.3 Interaction Modelling

Interaction modelling will play a crucial role in Chapters 4 and 5 of this thesis, as those Chapters will use Deep Reinforcement Learning techniques and Game Theoretic models described in this Section. Understanding and correctly modelling social interactions in autonomous driving would allow AV technologies to correctly predict the dynamic evolution of the surrounding scene, as well as enable AV engineers to generate socially acceptable AV behaviours. Accurately predicting the future trajectories of surrounding traffic participants would enable a much-required degree of safety that would allow AV technology to become a reality. An AV behaviour that is not well understood by the surrounding traffic might cause the AV to become an outlier amongst the traffic participants, thereby increasing the risk of traffic accidents [124]. Besides, understanding the social implications of AV behaviour would allow the AV actions to influence surrounding traffic, for example, by showing early stopping behaviour to encourage pedestrian crossing [125].

Interaction modelling techniques are relevant to a huge variety of autonomous driving tasks, ranging from traffic forecasting to AV planning and decision-making.

As these techniques can be utilised across different task domains, we will focus on dividing existing interaction modelling techniques regardless of the specific driving task that they have been designed for. The term *model* used in the remainder of this Section can refer to a generic solution for any autonomous driving task. For example, it can be a deep-learning model used for traffic forecasting or an interactive controller whose parameters need to be tuned.

While there has been extensive research in autonomous driving that makes use of machine-learning and deep-learning based techniques [126], a distinction can be made between learning-based methods and model-based methods. In a learning-based approach, a model is learnt from an extensive dataset. This set of methods do not require any prior knowledge of the system. Data-driven methods are trained on a dataset of examples, and then they are used to make predictions or decisions. On the opposite side of the spectrum, model-based methods start with a theoretical understanding of the system. This a priori-knowledge is used to create a mathematical model of the system. Empirical data is then used to validate the model or adjust its parameters to minimize the discrepancy between the model predictions and the data.

The literature includes a broad selection of techniques used to model interactions. We also make a further distinction between methods that explicitly utilise cognitive features of the human mind which try to explain the rationale that explains human actions, and methods that only implicitly try to model interactions, trying to map environmental inputs to decisions/actions. For instance, game theoretic methods (see Section 3.3.3) take a more explicit approach by considering traffic participants as rational agents who actively consider each other's actions. On the other hand, as an example of non-cognitive approaches, social force methods offer a more empirical perspective, capturing the impact of one participant on another, without explicitly detailing the reasoning that explains the agent's behaviour during the interaction. We propose to distinguish existing modelling approaches based on whether they explicitly or implicitly model the interactions.

Based on these two criteria, we identify four major categories of interaction modelling, which we report in Figure 3.7:

1. **Learning-based Implicit Methods**: These types of methods rely on machine learning or deep learning techniques. The interactions are implicitly modelled, which means that the agent's behaviour cannot be explained by the model. The model only learns an input-output mapping from the data. Model learning can be facilitated by exploiting interactive model architectures [127]–[130]. In general, deep learning methods that use interaction-specific neural network architectures fall into this category.

   In this type of method, the aim is to learn a probabilistic generative model that predicts the agent's future actions a. The model is a probability distribution conditioned on the environment state $\mathbf{x}$, which includes the state of surrounding agents, and a set of learnable parameters θ.

$$a \sim p_\theta(a|\mathbf{x}) \tag{3.1}$$

2. **Learning-based Methods with Cognitive Features**: This set of methods relies on explicitly handcrafted interactive features that are used as inputs for a learning based system. This type of interactive features can include TTC, relative distance [131], looming and reflecting some cognitive process behind human reasoning. For example, in [132] an LSTM which utilizes the inter-vehicle interactions has been developed to classify surrounding vehicles' lane change intentions. The interaction features are composed of risk matrices which account for worst-case TTC with vehicles in surrounding lanes and relative distance. Graph Convolutional Networks also fall into this category, as interaction features can be explicitly modelled in the adjacency matrix of the graph [133], [134].

   In this type of method, the aim is to learn a probabilistic generative model that predicts the agent's future actions a, similarly to 3.1. In this case, the probability distribution can be conditioned on the environment state $\mathbf{x}$ and on explicitly handcrafted interactive features $I(\mathbf{x})$, which have the purpose of facilitating learning.

$$a \sim p_\theta(a|\mathbf{x}, I(\mathbf{x}) \tag{3.2}$$

3. **Model-based Non-Cognitive Methods**: The modelling is non-cognitive in the sense that the interactions do not actively reason on the cognitive process that is behind the agent's actions. Methods of this group include SF [24] and potential fields. The interactions are described by potential functions (or SF) which contain a set of learnable parameters, which can be fit from empirical data. Another set of methods include driver risk fields, which are based on the hypothesis that the driver behaviour emerges from a risk-based field [43], [135]. The advantage of model-based implicit methods is that they can be easily interpreted and they can embed domain knowledge, such as traffic regulations and scene context. Some models define a potential field and define the agent's action as proportional to the gradient of such field:

$$a \propto \partial \mathbf{F}(\mathbf{x}) \tag{3.3}$$

   Otherwise, the forces can be modelled directly so that the gradient operation is not required $a \propto \mathbf{F}(\mathbf{x})$.

4. **Model-based Cognitive Methods**: Model-based cognitive methods describe the reasoning behind human's decision-making. We can distinguish between two main sets of methods: utility maximisation models and cognitive models.

   In utility maximisation methods, humans are modelled as optimizers that select their actions so as to maximise their future utility.

$$a = \arg\max_a U(a, \mathbf{x}) \tag{3.4}$$

These methods include game theory and Markov Decision Processes (MDPs). In Game theoretic approaches agents are modelled as players competing or cooperating with each other, thereby taking into account how they react to each other [136], [137]. The framework of game theory offers a transparent and clear-cut solution for modelling the dynamic interactions among human drivers, allowing for an understandable explanation of the decision process. However, it still remains hard to satisfy computational tractability as this approach does not scale well with an increasing number of agents. Another possible solution is to model human behaviour as an agent of an MDP, which provides an excellent framework to model decision-making in scenarios where results are influenced by both chance and the decisions made by a decision-maker. Solutions to MDPs can be found with learning based methods, e.g. DRL algorithms or Monte Carlo Tree Search [138], or with dynamic programming techniques [139].

The second set of methods aims to capture behavioural motivations behind agent's actions with psychological cognitive processes. This set of methods can include:

- **Stimulus-response** models [58], where driver or pedestrian actions are determined, for example, on visual stimuli in the retina;
- **Evidence accumulation** [85], where decisions are described as a result of accumulated evidence;
- **Theory of mind**, which suggests that humans use their understanding of others' thoughts and behaviours to make decisions. By predicting others' actions and inferring their knowledge, humans can drive effectively and safely [140], [141].

$$a = f_{cognitive}(\mathbf{x}) \tag{3.5}$$

In the next sections, we will analyse in greater detail each of these classes of interaction modelling. In particular, Cognitive and Non-Cognitive Learning based methods will be discussed in Section 3.3.1. Model-based cognitive methods have already been thoroughly discussed in Section 3.2, where we included Social Force and Potential Fields, Driver Risk Field models, Theory of Mind, Stimulus-Response and Evidence Accumulation models. Section 3.3.2 will include Utility-Based methods, which comprise MDPs (Section 3.3.2) and Game Theory (Section 3.3.3).

3.3.1 Learning Based Methods

Machine Learning (ML) methods are being used in autonomous driving for a variety of tasks, including object detection [142], scene understanding [143], path planning and control [23]. By learning from large amounts of data, ML methods can learn

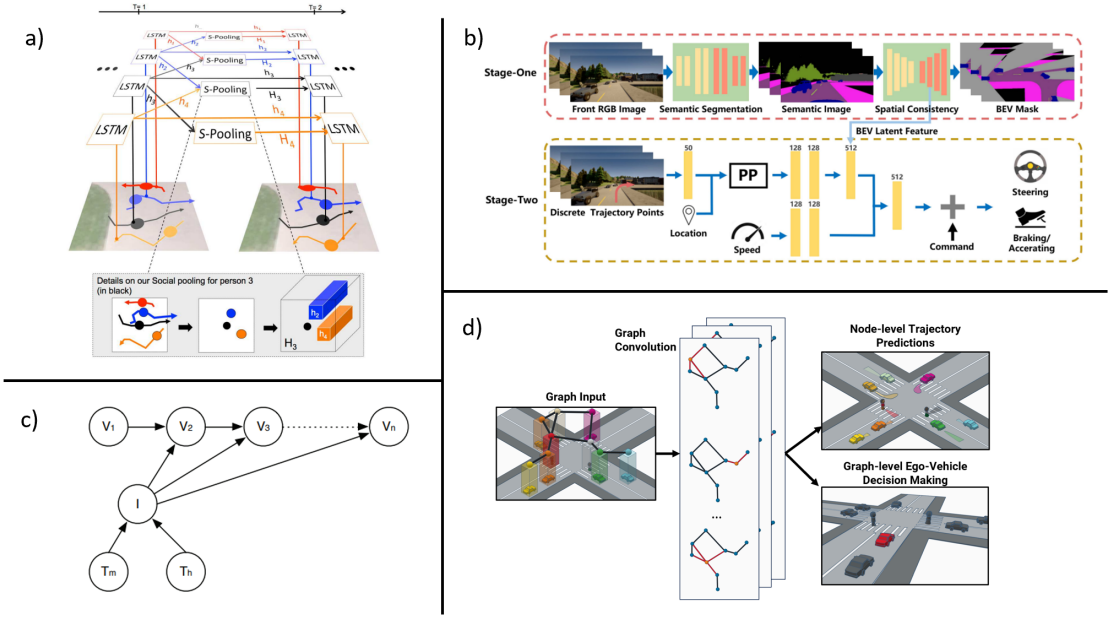


Figure 3.8: Overview figure of deep learning methods in interaction-aware tasks. a) social-pooling operation in [127], b) end-to-end imitation learning network in [145], c) probabilistic graphical model in [146], d) GCNs can be used for both node-level predictions of surrounding agents behaviour as well as ego-vehicle motion generation (graph-level output).

to make decisions that are more accurate and efficient than those made by humans [17]. This Section will comprise both implicit and explicit learning based methods identified in the previous Section and give a more detailed view of relevant papers. An overview of some Learning-based methods is shown in Fig. 3.8.

Thanks to recent improvements in neural networks learning representations, it is now possible to use *end-to-end* driving approaches that take as input the raw sensor readings to output control commands, such as steering and throttle to solve path-planning and control problems [144].

There are two main approaches to end-to-end self-driving for planning and control tasks:

- **Imitation Learning**: in which an agent learns to mimic the behaviour of an expert [147]–[149].
- **Deep Reinforcement Learning (DRL)**: in which an agent tries to learn how to act in a trial-and-error process that typically takes place is a simulated environment.

Different neural network architectures can be employed with Imitation Learning. In [148] interactive features are learned by means of a Graph Attention Network (GAT).

The input to this network consists of surrounding agents kinematic information as well as a feature vector that encodes scene representation coming from a Bird's Eye View. The model is trained on synthetic data generated by an expert driver in CARLA simulator. Imitation learning methods tend to work really well in scenarios that are similar to the training scenarios but typically fail when the scenarios diverge from the training distribution. Algorithms like Dataset Aggregation (DAgger) [150] can improve the performances of imitation learning policies by augmenting the initial training dataset with human-labelled data for unseen situations. However, asking an expert to label new training samples can be expensive or unfeasible. This problem is called distribution mismatch. To aid against this type of problem, an initial policy learned with imitation learning can serve as a starting point for DRL algorithms. Since the main framework for describing DRL methods is the MDP with utility-based agent, these methods will be analysed in greater detail in Section 3.3.2.

It important to note that it is challenging to learn the entirety of the driving task from high-dimensional raw sensory data (e.g. LiDAR point clouds, camera images) as this involves learning perception and decision-making at the same time. In most of the works, the *how-to-act* learning process assumes that a scene representation is available to the motion-planning and decision-making module. This actually requires splitting the end-to-end driving into two main blocks, one in which the AV learns *how-to-see* and one in which it learns *how-to-act*.

In the context of scene understanding and motion prediction, deep neural networks have been extensively used. [127] et al. proposed a social-pooling operation in their neural network architecture to account for surrounding neighbours in crowd motion prediction. Similarly, [151] made use of a star-topology network with max-pooling operation to account for interaction features in multi-agent forecasting. CIDNN [128] uses LSTM to track the movement of each pedestrian in a crowd and assigns a weight to each pedestrian's motion feature based on their proximity to the target pedestrian for location prediction. The study in [129] created a dataset and proposed a framework called VP-LSTM to predict the trajectories of vehicles and pedestrians together in crowded mixed scenes by exploiting different LSTM architectures for heterogeneous agents. A Generative Adversarial Network (GAN) is applied in [130] to sample plausible predictions for any agent in the scene. The shared feature of these methods is the usage of Recurrent Neural Networks that capture spatio-temporal interaction features in conjunction with pooling operations. The pooling operation allows to account for surrounding agents by mixing-up the hidden states extracted by the LSTMs. During the social-pooling operation, the hidden states of surrounding agents become features that are used to predict the current agent motion. Diffusion models are another set of deep-learning techniques with increasing popularity in modelling spatial-temporal trajectories, which can be used for predicting both pedestrian and car trajectories [152]. Other machine learning techniques that can be used to model interactions include Gaussian Processes. For example, [153] use LSTMs with social-aware recurrent Gaussian processes to model the complex transitions and uncertainties of agents in a

crowd.

Graph Convolutional Networks (GCNs) have been widely used in trajectory prediction tasks with interacting agents. In these methods, the road structure is represented as a graph, with each node representing a traffic participant. Each node can carry information such as the traffic participant's class (car, truck, pedestrian, etc.), its location, or speed. Explicit interaction can be modelled in the Adjacency Matrix of the graph, whereas the implicit part consists of the graph convolutional layers. GCNs are widely used in traffic forecasting [154]–[157], and have also been recently used in motion planning [158]–[161], especially in combination with DRL.

Probabilistic graphical models, including Hidden Markov Models have been employed in autonomous driving [162], [163]. For path planning, learning-based techniques that have been used in autonomous driving include Monte Carlo Tree Search and RL, see Section 3.3.2.

3.3.2 Utility Based Methods

The studies in this Section are of particular interest for RQ1 introduced in Chapter 1. A utility-based agent makes decisions based on a utility function. The utility function is a mathematical expression that assigns a value to each possible state of the world. The agent then chooses the action that leads to the state with the highest utility [164]. Utility-based agents are more complex than goal-based agents, which only consider whether or not a given state satisfies a goal. Utility-based agents can consider multiple goals and weigh them against each other. They can also consider the probability of different states occurring and the cost of taking different actions. For example, a utility-based agent could be used to decide which route to take to a destination, taking into account factors such as traffic, fuel efficiency, and cost.

We analyse two different utility-based methods in this Section: Markov Decision Processes (Section 3.3.2) and Game Theoretic Models (Section 3.3.3).

Markov Decision Processes

MDPs are a mathematical framework used to model decision-making problems where the outcomes are partly random and partly under the control of a decision-maker. The modelling framework for MDPs is illustrated in Fig. 3.9. Two main methods exist to solve MDPs: dynamic programming and reinforcement-learning [139]. Typically the latter set of methods are more used in autonomous driving, as they are more suitable for high-dimensional state spaces.

Reinforcement Learning

Reinforcement Learning makes use MDPs to model the environment and comprises a set of algorithms that learn policies that maximise the expected reward. In RL an agent learns to behave in an environment by trial and error.

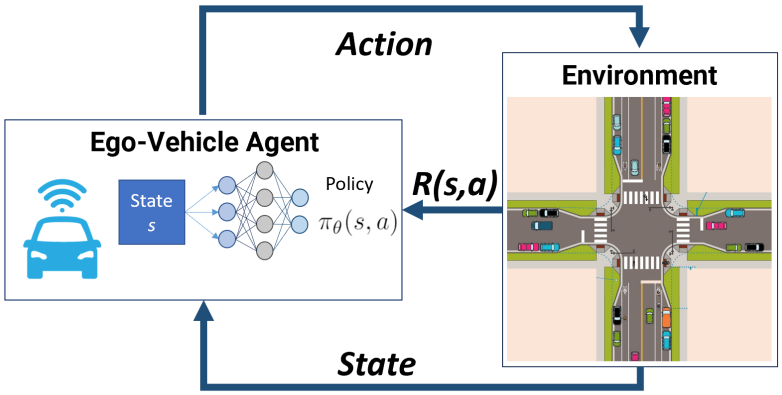


Figure 3.9: MDP framework. An agent takes an action that affects the environment state. The updated environment state is used to take the next action and the cycle repeats. The reward function is used to define the objective of the MDP, which is to maximize the expected cumulative reward over time.

The optimal policy for an MDP is the policy that maximizes the expected reward over time [139]. Dynamic programming can be used to find the optimal policy by iteratively computing the value of each state, starting from the terminal states and working back to the initial state. Dynamic programming can be a very effective way to solve RL problems, especially for problems with small state spaces. However, dynamic programming can be computationally expensive for problems with large state spaces, as in the Autonomous Driving case.

Traditional RL algorithms, such as Value Iteration or Policy Iteration algorithms [139] are not as well-suited for autonomous driving because they can be computationally expensive and sample-inefficient. RL algorithms need to interact with the environment for a long time to learn an optimal policy. They also need to be trained on a large dataset of experiences, which can be difficult to collect.

DRL is a type of RL that uses deep learning to learn from past experience, i.e. combines RL with Deep Neural Networks. DRL algorithms can be more sample-efficient and scalable than dynamic programming algorithms, but they can also be more complex and difficult to train. This field became more and more popular after Deepmind's groundbreaking articles [16], [17] in which an agent trained with DRL achieves super-human performances in video games. Agents trained with DRL have also been capable of exceeding human-level performances in continuous control [19] and robotics [20], [21], [165]. For a more detailed survey on DRL applications to autonomous driving, please refer to [23].

We will classify DRL solutions in autonomous driving based on the scenario used, the state space representation, the action space, and the algorithm used.

Since it is a very challenging task to map low-level features, i.e. LiDAR sensor data or raw camera images, into car actions, DRL methods assume that a perception

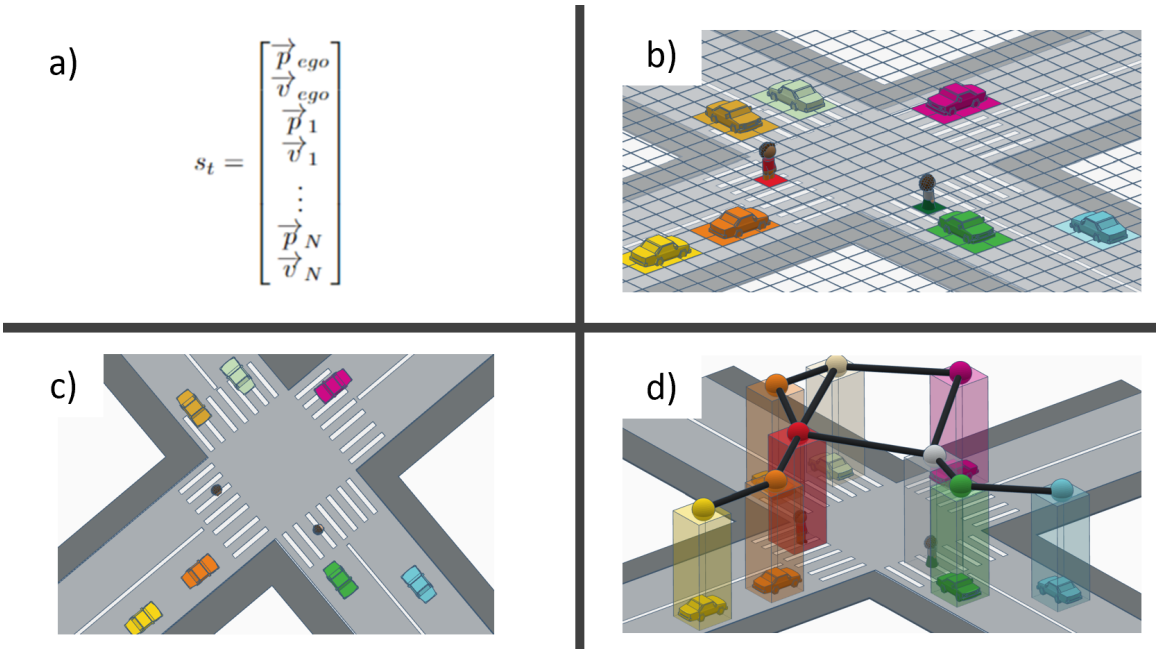


Figure 3.10: Illustration of state representations typically used in AD. a) vector representation, b) grid-based, c) Bird's Eye View, d) graph.

module first elaborates the surrounding environment into a high-level representation (Fig. 3.2) consisting of segmentation and obstacles/other agents identification. Typical state representations used in DRL include, see Figure 3.10:

- *Vector based representation*: in this type of representation information regarding surrounding vehicles, such as position and velocity, is included in a vector of fixed length [166];
- *Bird's Eye View (BEV) Image*: a 2D image representation of the environment surrounding the ego-vehicle from a top-down perspective [167];
- *Occupancy grid representation*: similar to a BEV image, it a it is a 2D discrete representation of the environment that is surrounding the ego-vehicle. It is a 2D or 3D grid of cells, each of which is assigned a probability of being occupied by an obstacle, as well as segmentation information regarding what type of entity is occupying the cell [168], [169].
- *Graph representation*: it is a way of representing the state of the environment around an AV as a graph. The nodes in the graph represent objects in the environment, such as vehicles, pedestrians, and traffic lights. The edges in the graph represent the relationships between objects, such as proximity or potential for collision. Graph representations are compact and efficient and are a promising approach to representing the state of the environment [160], [161].

Vector-based representation offers a compact and efficient representation of objects, at the expense of limiting traffic information to a subset of fixed dimensions of surrounding vehicles. BEV images and occupancy grids offer a simple way to represent the environment with fixed and can be easily updated, however, they can be inaccurate in environments with high clutter or uncertainty. Graph representation can easily represent the relationships between objects and is compact. On the other hand, it can be complex and computationally expensive to update the graph as the number of surrounding agents increases.

The action space can be continuous or discrete. Continuous actions usually include the ego-vehicle's longitudinal acceleration and steering angle. For example, in [170] the actions consist of the vehicle's jerk and angular velocity. Discrete actions usually depend on the specific task being solved. For example, in a lane change scenario, discrete actions include left-lane change, keeping the current lane, or right-lane change. Once a discrete action is selected, a lower-level controller regulates the steering and acceleration of the vehicle to execute such manoeuvre [169], [171]. There is a plethora of DRL algorithms to solve the problem, and each of them is suitable for a specific combination of actions and state representation. DRL environments are considered interactive, as the DRL agent will learn an optimal policy by experiencing interactions with simulated surrounding agents. Typically, DRL is performed in simulated environments, for safety reasons [23].

Whilst most DRL papers focus on vehicle-only traffic scenes, the number of papers that deal with mixed traffic scenarios or with vehicle-pedestrian interaction is more limited. Some works exist in the mobile robots crowd navigation. In [172] DRL is used to navigate a robot in a crowd in a multi-agent setting. In [173], the model in [172] is improved by using attention-based neural networks and social pooling. An autonomous braking system was developed in [13] with a DQN agent. The authors implement a Trauma Memory which is used to sample from collision scenarios, in a way similar to Prioritized Experience Replay (PER) [174]. In [12] a DQN agent is trained to avoid collisions with a crossing pedestrian and is further used to develop an ADAS system to aid drivers in pedestrian collision avoidance scenarios. Deshpande et al. [169], [175] used a grid-state representation with four layers.

One of the main challenges of DRL methods is how they can be deployed in the real world. Some research deploys a DRL model directly into the real world. In [176] a DRL policy based on attention mechanism is developed to handle unsignalized intersections. The authors show how the policy can deal with real-world scenarios by deploying directly into a vehicle model with any further fine-tuning. A field of machine learning, namely transfer learning is currently being explored to transfer knowledge from the simulated to the real world. Two main techniques include domain adaptation and domain randomization [177]. With domain randomization, we try to have a big enough training data so that it covers the real world as a special case [178]. With domain adaptation [179], the aim is to learn from a source distribution a model that performs well on a target distribution. Given the problems related to transferring

AV from simulated to the real world (safety and ethical issues) this area is an open research field. Another issue related to DRL is that the learning-based strategy has high training costs and is difficult to achieve semantic interpretation. Recently, some researchers focus on interpretable learning algorithms and lifelong learning algorithms to solve the above shortcomings [180].

Multi Agent Reinforcement Learning

The studies in this Section are of particular interest for RQ3 introduced in Chapter 1. When multiple RL agents are being deployed into the real world and interact with each other the problem becomes Multi-Agent Reinforcement Learning (MARL). This can be the example of Connected Autonomous Vehicles [158], where an optimal policy that manages all connected autonomous vehicles must be found. One of the main assumptions in DRL is that the environment is time-invariant. However, when multiple agents are learning at the same time, this hypothesis no longer holds and there might be training instabilities.

In order to deal with multi-agent systems multiple approaches are possible. The first approach is to have a centralised controller that manages the entire fleet. By increasing the state dimension to include all vehicles and having a joint action vector, the problem can again become a single-agent problem [181]. The drawback is the increased dimensionality of the state and action spaces, which can make learning more complex. Recently, graph-based representation has been employed to overcome the curse of dimensionality of the problem [181].

Another approach, which takes inspiration from level-k game theory is to have a single DRL learner but replace some of the surrounding agents with previous copies of itself [171]. This technique is similar to self-play, which is used in competitive DRL scenarios [111]. Finally, the last approach is to formulate the problem with a MARL approach, where multiple learners in parallel. A multi-agent deep deterministic policy gradient (MADDPG) method is proposed in [182], which learns a separate centralized critic for each agent, allowing each agent to have different reward functions. See [183] for an extensive review on MARL. Other applications of MARL in autonomous driving can be found in [158], [167], [168], [184].

3.3.3 Game Theoretic Models

The studies in this Section are of particular interest for RQ1 and RQ3 introduced in Chapter 1. In autonomous driving, the ego-vehicle must make decisions in an environment with multiple interacting agents. The actions of the ego-vehicle influence the behavior of surrounding vehicles, and vice versa, making the environment highly interactive. Game theory provides a framework for modeling these strategic interactions between rational agents [186]. Traditionally used in economics and political science, game theory has recently been applied to autonomous driving, particularly through dynamic non-cooperative game theory, which is relevant when multiple de-

Table 3.3: DRL overview table

Research	Scenario	Observation	Action	Reward	DRL Alg	Network	Simulator
[12]	Pedestrian Collision Avoidance	Continuous (x_p, y_p, v)	Discrete (break, keep, change lane)	coll, smooth, succ	DQN	MLP	PreScan
[166]	Highway Navigation	Continuous (x, y, v) + 8 surr. veh.	Discrete (LLC, RLC, KEEP) or (LLC, RLC, 4*ACC)	coll, near coll, lc, speed	DQN	CNN	Custom
[13]	Pedestrian Collision Avoidance	Continuous (x_p, y_p, v)	Discrete (ACC*4)	coll, early break penalty	DQN	MLP	PreScan
[175]	Pedestrian Collision Avoidance	Grid based representation	Discrete (ACC*4)	speed, coll, near coll	DQN	CNN +LSTM	Custom
[171]	Intersection	Continuous (v, path) + 4*surr. veh.	Discrete (5*ACC)	coll, near coll, acc, succ	D3QN PER	MLP	SUMO
[170]	Dense Traffic Lane Change	Grid	Continuous (jerk, ω)	coll, speed, jerk, time	PPO	CNN	Custom
[161]	Lane Change	Graph (p_{rel}, v_{rel}, lane index)	Discrete (LLC, RLC, KEEP)	speed, lc, coll	DQN	GCN	SUMO
[160]	Highway Lane Change	Graph (p_{rel}, v_{rel})	Continuous (ω, a)	coll, goal, vel, acc	PPO	GCN	BARK
[185]	Intersection, Roundabout	k-nearest, (p_{rel}, v_{rel}, θ)	Discrete	distance, coll	H-CtRL DDQN	MLP	Custom
[180]	Town Navigation	Lidar, Camera	Continuous (a, δ)	coll, speed acc, time, speed limit, out lane	Proposed	CNN	CARLA
[173]	Crowd-navigation	k-nearest (p_{real}, v_{rel}))	Discrete 80 (5*ACC, 16*angles)	coll, goal, proximity	Deep V-learning	Social-Attentive	Custom
[172]	Crowd-navigation	k-nearest (p_{rel}, v_{rel}, d, r	Discrete 11 (angle-speed comb.)	coll, goal, proximity	GA3C-CADRL	LSTM	Custom
[184]	MARL Intersection, Roundabout	all within range	-	time, speed, coll, front-car distance	Double DQN	Attention	Custom
[158]	MARL Highway Navigation	(p_{rel}, v_{rel}, lane, intention)	Discrete 3 (LLC, RLC, keep)	goal, speed, coll., lc	Graph Q (Proposed)	GCN	SUMO
[168]	MARL Merge	Cell grid	Discrete (ACC*5)	goal, coll, flow	Curriculum	MLP	Custom
[167]	MARL Connected AD	RGB BEV Image	Discrete 9 Combinations of (Brake, Steer, Throttle)	goal, speed, coll, lc	IMPALA	CNN	CARLA
[181]	MARL Connected AD	Graph (s, v, d)	Continous (a)	speed, action, idle, proximity, coll	TD3	GCN	Highway-env Open Source

cisions occur over time and agents pursue their interests independently, often with conflicting goals [10].

In traffic scenarios, all agents continually influence each other, balancing progress toward their destinations, collision avoidance, and compliance with traffic rules. Although originally designed for static games, game theory has been extended to dynamic games, including both discrete and continuous time formats. This approach extends optimal control theory to multi-agent settings, where optimal control is a special case involving a single player, with similar mathematical formulations [186].

Game theory, under the assumption that each player acts optimally, focuses on equilibrium solutions, particularly in trajectory games for autonomous driving. Depending on the information available to agents, dynamic games are categorized as open-loop (only initial state known) or feedback games (current state known). Although feedback games better represent the autonomous driving setting, open-loop solutions are simpler to compute and often provide reasonable approximations. Common equilibria used in autonomous driving include Open-Loop Nash, Open-Loop Stackelberg, Closed-Loop Nash, and Closed-Loop Stackelberg equilibria [186].

In Stackelberg competition, a leader moves first, and followers react sequentially, allowing higher-precedence players to anticipate others' responses. This approach has been applied in AV-human interaction models, where the AV assumes indirect control over human actions [10], although this can oversimplify complex interactions.

Generalized equilibria account for constraints like collision avoidance [207]. While open-loop Nash equilibria are computationally simpler, they lack the ability to model direct influence between agents. Solutions involving Stackelberg formulations and bimatrix games have been proposed for more complex scenarios, such as drone racing and autonomous vehicle interactions [191], [195].

Game-theoretic methods face challenges such as high computational complexity, the assumption of rationality, and the stochastic behavior of agents, which complicates solution finding. However, they effectively capture the interdependence of actions. To manage complexity, approaches like hierarchical game-theoretic planning, level-k reasoning, and iterative methods are employed [189], [194], [206]. Other methods, such as Iterative Best Response and Nash equilibrium reformulations, are also used to solve these problems [32], [188].

Further advancements include addressing uncertainties in agent intentions using models like POMDPs and constructing multiple hypotheses about agents' objectives and constraints [193], [203].

Table 3.4: Game Theory Models for Decision Making in Various Scenarios

Research	Scenario	Game Theory Model	Agents	Action
[187]	Lane Change	Nash Equilibrium	Up to 4	Both discrete (Lane change) and continuous a
[188]	Autonomous Racing	Nash Equilibrium	2	Continuous
[189], [190]	Intersection	Nash Equilibrium	3	Continuous (ω and a)
[191]	Drone Racing	Nash Equilibrium	2	Continuous (ω and v)
[192]	Autonomous Racing	Nash Equilibrium	2	Continuous (a and δ)
[193]	Autonomous Racing	Nash Equilibrium	2	Continuous (a and δ)
[32]	Merging	Nash Equilibrium, SVO	2	Continuous (a and δ)
[194]	Highway, Lane Change	Nash Equilibrium, DRL	2	Discrete (a and lane change)
[195]	Autonomous Racing	Nash Eq. and Stackelberg Eq.	2	Discrete trajectories
[10]	Highway navigation, Intersection	Stackelberg Eq.	2	Continuous (a and δ)
[196]	Lane Change	Stackelberg Eq.	3	Both discrete and continuous
[197]	Lane Change	Stackelberg Eq.	2	Discrete (Lane change)
[136]	Roundabout	Stackelberg Eq.	2	Continuous
[198]	Truck Platooning	Hierarchical Stackelberg Eq.	2	Continuous (a and δ)
[199]	Lane Change	Stackelberg Equilibrium	2	Discrete (Lane change)
[200]	Highway navigation	Hierarchical Feedback Stackelberg Eq.	2+	Continuous
[201]	Highway navigation, Merge	Game Tree	2	Discrete
[202]	Merge	Game Tree	2	Discrete
[203]	Highway navigation	Generalised Feedback Nash Eq.	2+	Continuous
[137]	Left Turn	Generalized Nash Eq., SVO	2	Continuous (a and δ)
[204]	Drone Racing	Generalized Nash Eq.	6	Continuous (v)
[205]	Merge, Intersection	Generalized Nash Eq.	4	Continuous (a and ω)
[11]	Intersection	Level-k	2	Continuous (a and ω)
[206]	Roundabout	Level-k	2	Continuous (a and ω)

Chapter 4

Interaction-aware decision-making for AVs

Autonomous driving is a rapidly advancing technology with the potential to dramatically reduce traffic accidents. Despite substantial advancements in Autonomous Vehicle (AV) research, the global road death toll remains disturbingly high. A major challenge in autonomous driving is achieving collision-free navigation in complex, interactive environments where pedestrians are present. Traditional motion control algorithms tend to be overly cautious, leading to unpredictable driving behavior. These methods also struggle to adapt to diverse, real-world scenarios.

In this Chapter, we propose to solve the above issues by adopting a learning-based approach and by utilising the concept of Social Value Orientation (SVO) from Social Psychology [30], [31], [208] into the AV motion controller design. SVO is a value that quantifies how much a person values the welfare of the others compared to their own. We are witnessing an increasing number of publications seeking to utilise Deep Reinforcement Learning (DRL) to solve Autonomous Driving problems [185], [209]. In DRL, a motion controller is synthesised through trial-and-error interactions with simulated environment without the need to manually handcraft an AV decision-making policy, making maintenance and development simple. For a more comprehensive review on the topic see [23]. Reinforcement Learning applications in the field of autonomous driving mainly focus on navigation amongst other vehicles, and the problem of pedestrian collision avoidance in structured environments is a less studied one. Besides, studies in the field of pedestrian collision avoidance mainly focus on unstructured scenarios, where the vehicle and the pedestrians share a common space. In our study, we focus on a typical lane-crossing scenario, where the pedestrian and the vehicle usually occupy separate regions of space, i.e. the road and the pavement, and interact only at some predefined regions, for example, at crossings. To the best of our knowledge, this is the first time that SVO has been used to shape the reward function in RL to the problem of pedestrian collision avoidance in structured scenarios.

Our innovative approach merges DRL with SVO to develop autonomous vehi-

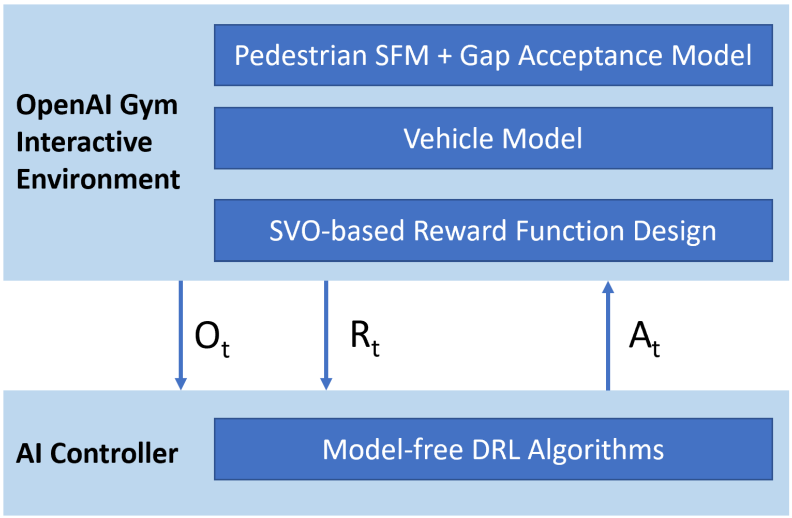


Figure 4.1: Technical framework used. O_t, R_t, A_t represents reinforcement learning observations, reward, and actions respectively.

cle (AV) motion controllers that consider the comfort and well-being of nearby road users. While traditional DRL methods concentrate solely on the objectives of the ego-vehicle, disregarding potential negative consequences for other vehicles, our method introduces a pivotal shift. Specifically, we redefine the reward function in DRL by incorporating SVO, thereby considering the impact of AV actions on surrounding road users, with a particular emphasis on pedestrians. In our research, we simulate a common scenario involving a single pedestrian and train a series of DRL agents with diverse SVO values to enhance the AV's interaction with its environment. Our proposed framework is shown in Fig. 4.1. The work described in this Chapter constitutes papers 1 and 2 ([210], [211]) listed in the publication Section of this thesis. In the first study [210], SVO is introduced in the DRL in a simplified environment with a pedestrian model from [212].

In the second study, an interactive pedestrian model is created that captures the dynamic relationship between vehicle motion and pedestrian decisions. Since the pedestrian model will be extensively used to train our DRL agent, we need it to possess three main characteristics: firstly, we require it to be computationally efficient to avoid bottlenecks during training; secondly we need it to be realistic; and finally we need the pedestrian to actively reason about the AV's decision, so as to achieve an interactive behaviour that can be learnt and exploited by the AV agent. In this Chapter, we propose a novel pedestrian model that integrates the concept of situational awareness into the Social Force Model framework to achieve an interactive behaviour that explicitly reasons about the AV's actions.

Several works have studied the vehicle-pedestrian interaction in crossing scenarios. A comprehensive review on pedestrian models in autonomous driving can be found in [213] and [214]. Gap acceptance is a major factor that influences pedes-

trian decision at intersections. Gap acceptance models have been used to describe the probabilities of pedestrians crossing in a certain gap between vehicles [58], [215]–[217]. These models are used to describe the pedestrian crossing probability but they do not model the trajectories that the pedestrian will follow. Markkula *et al.* [25] introduced the concept of situational awareness in the pedestrian crossing modelling. In their model, the authors describe pedestrian road crossing decision as the result of a number of perceptual decisions concerning the available gap. A limitation in pure gap-acceptance models is the assumption that once a pedestrian initiates crossing, they will follow a constant speed velocity profile.

On the other hand, Social Force Models [26] of pedestrian behaviour describe collective behaviours by modelling how each individual interacts with other. The idea behind this model is that the influence of surrounding agents on the pedestrian motion can be modelled with forces that measure for the internal motivations of the individuals to perform certain movements. This model was originally designed for simulating crowd dynamics but has been extended with the effect of vehicles on pedestrians [97], [212], which makes them suitable for mixed scenarios containing both vehicles and pedestrians. Existing works [218], [219] in pedestrian simulation combined social force models with a rule based approach for pedestrian crossing simulation. These models however mainly focus on situations in which pedestrians are in front of the vehicle. In our DRL setting, especially when the vehicle policy is not yet trained, episodes in which the vehicle and the pedestrian are next to each other will be present, which is why we extend the pedestrian model to such scenarios. Secondly, we add a temporal aspect to the decision making process, by including situational awareness in the pedestrian decision making.

This model is evaluated against real-world data and the previous pedestrian model. We also compare different model-free DRL algorithms to train AV agents, which learn interaction patterns with pedestrians, indirectly influencing their behavior. This results in controllers exhibiting human-like behavior, offering a wide spectrum of driving styles based on the chosen SVO value. We conduct a comprehensive set of experiments to assess the impact of SVO integration and model performance across various scenarios, moving us closer to safer and socially aware autonomous vehicles. This chapter has the following contributions:

1. We introduce the concept of SVO in the RL reward function design to obtain control policies that take the pedestrian goal into account, achieving egoistic or pro-social AV behaviour;

2. We demonstrate how the introduction of SVO into the DRL Reward Function design influences the ego-vehicle strategies, achieving behaviours that range from egoistic to pro-social, without affecting pedestrian safety;

3. We show a successful application of a state of the art RL algorithms, namely the Soft-Actor Critic (SAC), and Proximal Policy Optimisation (PPO), to the pedestrian collision avoidance problem.

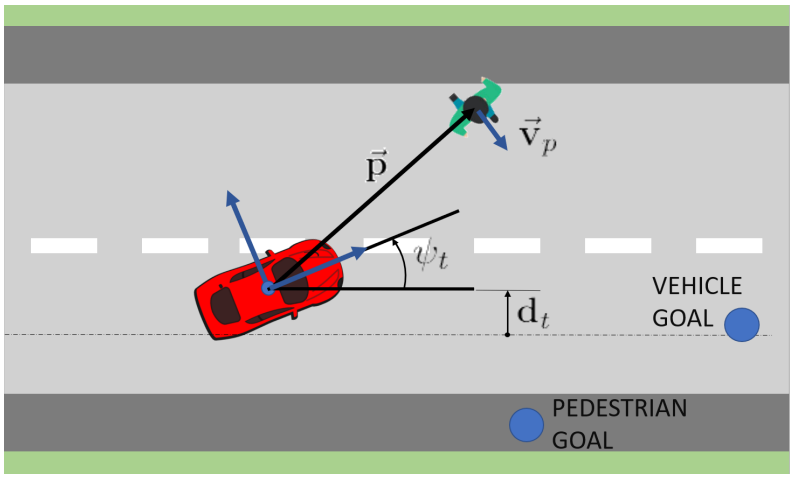


Figure 4.2: Scenario illustration. The vehicle measures the pedestrian position and velocity and adjusts its longitudinal acceleration to balance time efficiency and pedestrian safety.

4. We introduce a novel pedestrian simulation model that combines gap-acceptance methods with Social Force Models to model the pedestrian crossing behaviour;

5. We validate that our RL model is capable of handling the added complexity introduced by our more realistic pedestrian model that actively reasons about the AV's actions and conduct a comparative analysis of two model-free DRL algorithms applied to our problem.

4.1 Methodology

4.1.1 Social Value Orientation

In social psychology, Social Value Orientation (SVO) is a value that describes how much a person values other people's welfare in relation to their own. With SVO theory in mind, we can model each individual as a decision-making agent that maximises their own utility function. Such a utility function can be expressed as a combination of the ego agent's utility U_{self} and other agents' utility U_{other}:

$$U_{total} = \cos(\varphi)U_{self} + \sin(\varphi)U_{other} \tag{4.1}$$

where φ is the SVO. It is an angle, whose value affects the weights of the two utility terms, and therefore the balance between the selfish reward and the altruistic reward. As we can see in Figure 4.3, we can characterise the personality of each individual with the SVO value. For example, an SVO value of 90° corresponds to fully altruistic behaviour, whereas an SVO value of 0° corresponds to an individualistic agent. In our

work, we focused on SVO values between 0° and 90°, as we want the AV to exhibit pro-social behaviour and yield to the pedestrian if necessary to avoid dangerous situations.

SVO has been previously used to design controllers in a game-theoretic setting [32], but this demands long complex computations to solve for a Nash equilibrium. In our work, we try to mitigate the computational cost of the optimisation problem by using SVO in the RL framework, thereby moving the computational cost from execution time to training time, in a learning-based fashion. We integrate the SVO concept directly in the MDP model formulation, by constructing a reward function that is composed of two terms, one that models the car's own objective U_{self} and one that models the pedestrian's objective U_{other}. As model-free RL algorithms are computationally less complex and do not require an accurate representation of the environment to be effective, we choose the SAC algorithm, which has proven to be very effective for autonomous car traffic navigation [220]. The SAC has also the advantage of using a continuous action space, which is more suitable for our problem. To generate realistic and interactive experience at training and test time, we use social forces to model the pedestrian's behaviour. We adopted a model similar to the one used in [212], where we considered a single vehicle and neglected the interaction with other pedestrians. We include an awareness probability for the pedestrian to improve robustness to collisions and avoid overfitting of the pedestrian behaviour. This way, the trained policy will be able to deal with dangerous situations where the pedestrian starts crossing without seeing the vehicle. In the next section, we introduce the state and action spaces of the MDP and in section 4.1.3 we describe the social reward function design with SVO.

4.1.2 MDP Formulation

We design a Deep Reinforcement Learning environment in which the Autonomous Vehicle (the RL agent) interacts with the pedestrian. We let the AV learn its behavioural policy by experiencing interactions with the pedestrian model we developed. We use two different DRL algorithms, SAC and PPO, to train two sets of policies and compare their performances. In our first study [210], the SAC algorithm is used in combination with a predefined state-of-the-art pedestrian model [212]. In our second study [211], we developed a novel pedestrian model to be used in this crossing scenario and the performance of two different model-free DRL algorithms are compared. We will now describe how we model the pedestrian collision-avoidance problem as a Markov Decision Process (MDP) that can be used to train DRL AV agents.

State Space

In our model, we focused on a scenario consisting of a straight lane and a single pedestrian, see Figure 4.2. This environment serves as a critical component in enabling the AV (acting as the RL agent) to develop its behavioural policy through real-time interactions with a pedestrian model. We assume the ego-vehicle can access a ref-

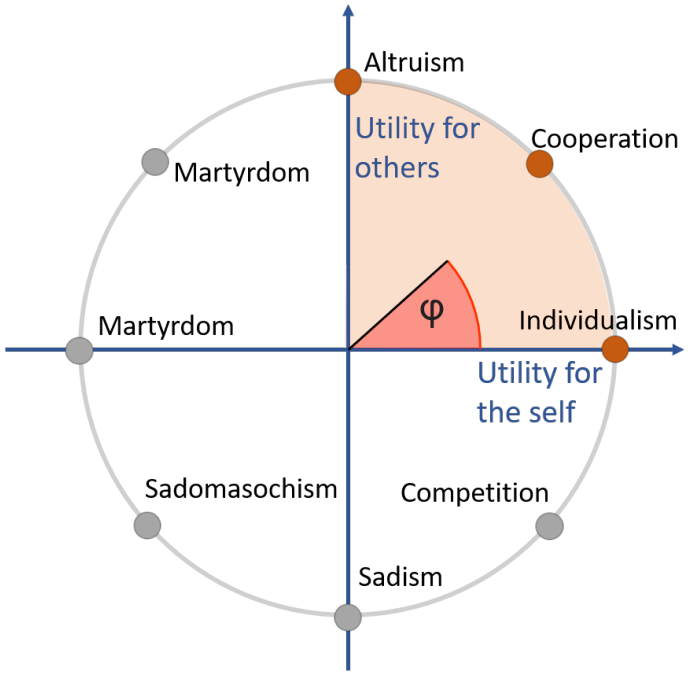


Figure 4.3: Social Value Orientation ring. The SVO value φ affects the behaviour of the ego-vehicle.

erence path computed by a Path-Planning module using the GPS and global map information and that it is able to locate itself with respect to this reference path. The purpose of the control algorithm is to adapt the vehicle current trajectory to the reference track based on the pedestrian's behaviour. As for pedestrian detection, we assume that the vehicle is equipped with a LiDAR, which measures the pedestrian position relative to the vehicle. Therefore, we make the assumption that the state space available to the ego-vehicle consists of:

- the offset d_t of the vehicle centre of gravity from the vehicle reference path;
- the vehicle orientation relative to the reference path direction ψ_t;
- the vehicle longitudinal velocity v_t^{ego};
- the pedestrian position $\mathbf{r}_t^{ped}$ and velocity $\mathbf{v}_t^{ped}$ relative to the vehicle.

$$s_t = [d_t, \psi_t, v_t^{ego}, \mathbf{p}_t, \mathbf{v}_t^{ped}]^T \in \mathbb{R}^7 \tag{4.2}$$

In this work, we set the vehicle orientation ψ_t to 0, to study the vehicle's longitudinal motion.

Action Space

For the action space, we have adopted a continuous representation in the state space. Specifically, the action space is comprised of the AV's longitudinal acceleration denoted as a_t. This action, computed by the policy network, is normalized to fall within the interval $[-1, 1]$, and then rescaled to fit within the range of $[-0.3g, 0.3g]$, where g signifies the standard gravity of the Earth. This approach to the action space allows for a fine-grained control over the AV's acceleration.

4.1.3 Social Reward Function

We introduce the SVO concept directly inside the reward function. The integration of SVO into the reward function allows us to infuse a sense of social responsibility and consideration for the well-being of other road users. The fundamental structure of our reward function encompasses two distinct terms r_{car} and r_p, each of which plays a pivotal role in guiding the AV's behavior. We indicate the ego-vehicle's SVO value as φ, and weight each of the two distinct forms with the sine and cosine function of the ego-vehicle's SVO:

$$r(s_t, a_t) = \cos\varphi \cdot r_{car}(s_t, a_t) + \sin\varphi \cdot r_p(s_t, a_t) \tag{4.3}$$

Each term solves a specific function:

- **Ego-Vehicle Objective Term**: this term encapsulates the AV's own objectives and goals, aligning with traditional reward functions used in Deep Reinforcement Learning (DRL). It factors in elements such as reaching a destination efficiently and adhering to traffic rules, ensuring that the AV's autonomous decision-making caters to its primary objectives.
- **Pedestrian Social Term**: This novel addition introduces the SVO concept, and its role is to account for the comfort and well-being of other road users, particularly focusing on interactions with pedestrians and nearby vehicles. By integrating SVO directly into the reward function, we encourage the AV to make decisions that not only optimize its own objectives but also consider the impact of those decisions on the safety and comfort of surrounding individuals. This represents a significant departure from conventional DRL methods that tend to concentrate solely on the ego-vehicle's goals and neglect the broader societal implications of its actions.

The car's reward function is also a combination of multiple terms:

$$r_{car}(s_t, a_t) = r_c + r_g + r_v \tag{4.4}$$

The first term is a collision term r_c, the second term is a positive reward to achieve the goal r_g and r_v is a reward related to the vehicle speed. For avoiding collisions with

pedestrians, a penalty r_c of -30 is given to the car in case of collision and the episode is terminated. A positive reward r_g of $+30$ is given to the agent when it reaches the goal. These values have been set emprically by manual search. Finally the term r_v is a speed reward that encourages the car to reach the goal in the minimum amount of time. It is expressed as:

$$r_v = c_1 \mathbf{v_t} \cdot \hat{\mathbf{n}} = c_1 \cos \psi_t v_t \tag{4.5}$$

The term $\mathbf{v_t} \cdot \hat{\mathbf{n}}$ is the dot product between the vehicle velocity and a unit vector representing the reference path forward direction. This encourages the car to go as fast as possible within the speed limit and at the same time discourages the car from moving backwards.

The second term of equation 4.3 is used to capture the pedestrian's intentions and comfort in the AV's decision-making process. We assume a pedestrian crossing the road is attempting to reach their goal in the least amount of time possible, so we give a positive reward proportional to the pedestrian crossing speed to the AV when the pedestrian is crossing. Also, since we want our RL agent to behave pro-socially, i.e. yielding to the pedestrian if necessary, we give a positive reward only if the pedestrian is crossing in front of the vehicle. Since an AV stopping in close proximity of a pedestrian to let them cross could be potentially dangerous or could make the pedestrian feel unsafe, we weight the reward by a factor $\sigma\left(D_{pv}\right)$, where D_{pv} is the distance between the vehicle and the pedestrian. $\sigma\left(D_{pv}\right)$ is a sigmoid function that tends to 0 when D_{pv} tends to 0. If $\mathbf{v}_p$ is the pedestrian velocity, we can then express the pedestrian reward function as:

$$r_p = \begin{cases} k_p \sigma(D_{pv}) \vec{\boldsymbol{v}_p} \cdot \hat{\rho}, & \text{if } \textit{wants to cross} \quad \textbf{and} \quad x_p > x_v \\ 0, & \text{otherwise} \end{cases} \tag{4.6}$$

where k_p is a scaling coefficient for the pedestrian reward, $\vec{\boldsymbol{v}_p}$ is the pedestrian velocity,, $\hat{\rho}$ is a unit vector pointing from the pedestrian to their goal position, x_p and x_v are the pedestrian and vehicle positions along the x dimension (aligned with the road).

4.1.4 Situational Aware Pedestrian Model

We let the vehicle agent learn the best control action based on the experience of the dynamic interactions with a pedestrian in a simulated environment. Rather than treating the pedestrian as a mere moving obstacle, we emulate a real-world pedestrian motion with a social force-based model and add a social term in the reward function that is based on Social Value Orientation.

First, we develop a novel interactive pedestrian model that combines the concepts of situational awareness [25] and Social-Force [26] to determine the pedestrian trajectory under the vehicle influence. The vehicle motion affects the pedestrian decisions by indirectly altering the available time-gap to complete crossing and the social forces acting on the pedestrian. In turn, pedestrian motion serves as a cue for

the ego-vehicle controller, thereby mutually influencing each other. We evaluate our pedestrian model using a set of typical road scenarios and by comparing pedestrian motion statistics with real world data and a state-of-the-art pedestrian model. Secondly, agents trained with model-free DRL algorithms learn the interaction patterns with the pedestrian and exploit them to indirectly affect pedestrian motion. For instance, the vehicle learns the effect that its own acceleration on pedestrian's decisions, thereby hindering or favouring the pedestrian crossing. We demonstrate how our reward choice produces controllers that naturally exhibit human-like behaviour, with a plethora of different driving styles, ranging across a spectrum from aggressive to pro-social according to the choice of the SVO value. We conduct a set of qualitative and quantitative experiments aimed at evaluating the effect of SVO addition, and model performances under both nominal and high-risk scenarios. We modify the traditional social force based model by mixing it with a gap-acceptance model to simulate pedestrian crossing behaviours. In this way, we are able to obtain realistic trajectories due to the social force component while still maintaining the advantages of gap-acceptance models, i.e. the accurate description of crossing initiation.

Pedestrian's Motivation

We model the pedestrian situational awareness as a number that represents the pedestrian's willingness to cross the road, which we term *motivation*. Inspired by the work of [25], we model the pedestrian's motivation as a discrete time variable that quantifies the pedestrian's crossing willingness. The motivation takes into account environmental factors such as the AV's forward velocity v_v, the distance between the pedestrian and the vehicle D_{pv}, the lane width and the vehicle's acceleration perceived by the pedestrian a.

The motivation at any point in time $M(t)$ is a real value in the interval $[0, 1]$, with 1 indicating that the pedestrian wants to cross the road and 0 the opposite. In order to model the fact that the decision-making process is made over time, we apply a first order filter and update the motivation according to the following equation:

$$M(t+1) = \alpha M(t) + (1-\alpha)\hat{M}(t) \tag{4.7}$$

where $\hat{M}(t)$ is an innovation term that is computed according to the vehicle's position and actions and $M(t)$ is the motivation at the previous timestep.

The innovation term is computed as a logistic function:

$$\hat{M}(t) = \frac{1}{1 + e^{-(\boldsymbol{\psi}^T \mathbf{f} - \beta)}} \tag{4.8}$$

where $\mathbf{f}$ is a vector of features, $\boldsymbol{\psi}$ is a vector of weights, and β is a parameter. The vector of features combines the advantage time and the acceleration of the vehicle perceived by the pedestrian:

$$\mathbf{f} = [t_{adv}, a]^T \tag{4.9}$$

In particular, We define the advantage time t_{adv} as the difference between the time to collision and the time that the pedestrian needs to cross the road, considering their reaction time:

$$t_{adv} = \frac{D_{pv}}{v_v} - \frac{kL}{v_d} - t_r \tag{4.10}$$

where L is the lane width, k is a coefficient which is equal to 1.0 if the pedestrian initiates crossing on the same side as the vehicle's lane or 2.0 otherwise, indicating that the pedestrian has to travel only half the road width or the total road width. t_r is an additional time factor that takes into account pedestrian reaction time. The terms v_v and v_d represent the vehicle's speed and the pedestrian desired walking speed.

Navigational Force

The navigational force is a proportional controller that drives the pedestrian towards their goal, weighted by the current pedestrian motivation:

$$\vec{\mathbf{F}}_{nav}(t) = M(t) \cdot k_d \left(\vec{\mathbf{v}}(t) - \vec{\mathbf{v}}_d(t)\right) \tag{4.11}$$

The desired velocity $\vec{\mathbf{v}}_d(t)$ points at each timestep in the direction of the goal $\vec{\boldsymbol{g}}$ and has a magnitude equal to the pedestrian's preferred walking speed v_d:

$$\vec{\mathbf{v}}_d(t) = v_d \frac{(\vec{\boldsymbol{g}} - \vec{\boldsymbol{p}})}{\sqrt{\|\vec{\boldsymbol{g}} - \vec{\boldsymbol{p}}\|^2 + \varepsilon_n^2}} \tag{4.12}$$

where $\vec{\boldsymbol{p}}$ is the pedestrian's current position, and ε_n is a regularisation factor for the navigation term to avoid the problem of division by zero.

Vehicle Interaction

We modelled the vehicle influence on the pedestrian as a superposition of three different force fields. The first term affects the pedestrian trajectory so that they avoid collisions with the vehicle, the second term encourages walking around the vehicle when it has very low speed, and the last term pushes the pedestrian away from the front area of the vehicle if it is approaching with high speed. Since a pedestrian will avoid walking in the area in front of a vehicle approaching at high speed and will not initiate walking around it unless the vehicle speed is sufficiently low, we take this into account by introducing a velocity coefficient that blends the second and third term according to the vehicle's speed. In particular, we define the overall force field as:

$$\vec{\mathbf{F}}_{veh} = \vec{\mathbf{F}}_{shape} + k(v)\vec{\mathbf{F}}_{flow} + (1 - k(v))\vec{\mathbf{F}}_{speed} \tag{4.13}$$

where

$$k(v) = \frac{1}{1 + k_v v^2} \tag{4.14}$$

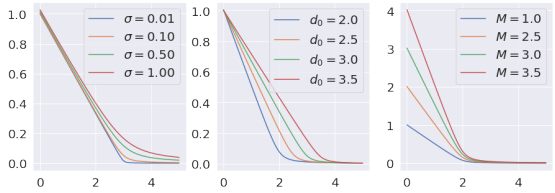


Figure 4.4: Linear decay with smoothing. Values are unitless.

The parameter $k(v)$ is used to obtain a linear combination of the fields $\vec{\mathbf{F}}_{flow}$ and $\vec{\mathbf{F}}_{speed}$, so that at lower velocities the former prevails, whereas at higher speeds the latter prevails. Let $\vec{\mathbf{p}} = [x, y]^T$ be the coordinates of a pedestrian in the vehicle local frame.

The shape of the fields $\vec{\mathbf{F}}_{shape}$ and $\vec{\mathbf{F}}_{flow}$ is shown in Fig. 4.5. We approximate the AV shape as an ellipsis for the sake of the repulsive force modelling, with semi-axes a and b, equal to half the vehicle length and width respectively.

We use a linear decay function with smoothing to model the influence of the vehicle shape on the pedestrian based on the distance d between the vehicle and the pedestrian, which is defined as:

$$h(d; A, d_0, \sigma) = \frac{A}{2d_0}\left(d_0 - d + \sqrt{(d_0 - d)^2 + \varepsilon}\right) \tag{4.15}$$

where A, d_0, and ε are parameters that determine the shape of the linear decay function and whose effects are shown in Fig. 4.4. We use an elliptical distance in accordance to the AV's shape approximation:

$$d = \sqrt{\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2} \tag{4.16}$$

where x and y are the pedestrian's coordinates relative to the vehicle.

The repulsive force direction is orthogonal to the vehicle's shape approximation and its magnitude depends on a linear decay with smoothing function of the elliptical distance. The denominator normalises the equation to get a unit vector in the desired

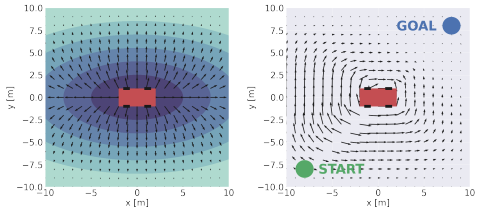


Figure 4.5: Shape field (left) and force field (right) representation. The flow field is shown with two randomly chosen start and goal positions.

direction:

$$\vec{\mathbf{F}}_{shape} = \frac{h(d; A_s, d_{0s}, \sigma_s)}{\sqrt{\left(\frac{2x}{a^2}\right)^2 + \left(\frac{2y}{b^2}\right)^2}} \left(\frac{2x}{a^2}\hat{\boldsymbol{\imath}} + \frac{2y}{b^2}\hat{\boldsymbol{\jmath}}\right) \tag{4.17}$$

The flow field encourages the pedestrian to walk around the vehicle. We introduce a coefficient $k_f(\vec{\boldsymbol{p}})$ which has two purposes. Its sign determines if the pedestrian walks around the vehicle clockwise or counterclockwise and it is decided by estimating the shortest path between the pedestrian current position and the goal position. The magnitude of the coefficient $k_f(\vec{\boldsymbol{p}})$ is one at the beginning of the trajectory and decreases to zero as the pedestrian is closer to the goal. This choice was made because the vehicle should not influence the pedestrian motion once the pedestrian has passed the vehicle and is moving further away from it. In symbols:

$$|k_f(\vec{\boldsymbol{p}})| = \begin{cases} 1.0 & \text{if } P < 0 \\ 0 & \text{if } P > \|\vec{\boldsymbol{g}} - \vec{\boldsymbol{p}}_0\| \\ \frac{\|\vec{\boldsymbol{g}} - \vec{\boldsymbol{p}_0}\| - P}{\|\vec{\boldsymbol{g}} - \vec{\boldsymbol{p}_0}\|} & otherwise \end{cases} \tag{4.18}$$

where P is the pedestrian progress towards their goal and $\vec{\boldsymbol{p}_0}$ is the pedestrian initial position:

$$P = \frac{(\vec{\boldsymbol{p}} - \vec{\boldsymbol{p}_0}) \cdot (\vec{\boldsymbol{g}} - \vec{\boldsymbol{p}_0})}{\|\vec{\boldsymbol{g}} - \vec{\boldsymbol{p}_0}\|} \tag{4.19}$$

We allow the flow term to have its own linear decay with smoothing parameters. Compared to the shape field, the flow field has a different power for the x and y terms in eq. 4.20 to make the pedestrian trajectory follow more realistic paths around the vehicle. The negative sign in the last term of equation 4.20 makes the field rotate around the vehicle. The pedestrian flow term can then be defined as:

$$\vec{\mathbf{F}}_{flow} = k_f(\vec{\boldsymbol{p}}) \frac{h(d; A_f, d_{0f}, \sigma_f)}{\sqrt{\left(\frac{-2y^3}{b}\right)^2 + \left(\frac{2x^3}{a}\right)^2}} \left(\frac{-2y^3}{b}\hat{\boldsymbol{\imath}} + \frac{2x^3}{a}\hat{\boldsymbol{\jmath}}\right) \tag{4.20}$$

The influence of the vehicle speed on the pedestrian motion is modelled with the force field $\vec{\mathbf{F}}_{speed}$, which follows an exponential decay:

$$\vec{\mathbf{F}}_{speed} = A \cdot \text{sign}(y) \exp\left(-\frac{x - a}{v\Delta T}\right) \exp\left(-\frac{y^2}{2\sigma_y^2}\right)\hat{\boldsymbol{\jmath}} \tag{4.21}$$

where A is a scaling coefficient, v_v represents the vehicle speed, ΔT is a time factor, and σ_y is a constant proportional to the lane width. The exponential decay is influenced by the vehicle speed, varying the length of the area that is influenced in front of the AV.

The overall pedestrian model is shown in algorithm 1.

Algorithm 1 Social Force–Motivation Pedestrian Model

Input: Pedestrian's position $\vec{\boldsymbol{p}} = [x, y]^T$, pedestrian's goal position $\vec{\boldsymbol{g}}$, vehicle speed v and acceleration a.
Output: Pedestrian's $\vec{\boldsymbol{a}}$.
Parameters: see Table 4.1.
Compute Pedestrian Speed:
Update Motivation:
$\hat{M}(t) \leftarrow \frac{1}{1+e^{-\boldsymbol{\psi}^T \mathbf{f}}}$
$M(t) \leftarrow \alpha M(t-1) + (1-\alpha)\hat{M}(t)$
Update Forces:
if $M(t) > \theta_f$:
$\vec{\mathbf{F}}_{nav}(t) \leftarrow M(t) \cdot k_d \left(\vec{\mathbf{v}}(t) - \vec{\mathbf{v}}_d(t)\right)$
else:
$\vec{\mathbf{F}}_{nav}(t) \leftarrow \mathbf{0}$
$\vec{\mathbf{F}}_{sh}(t), \vec{\mathbf{F}}_f(t), \vec{\mathbf{F}}_{sp}(t) \leftarrow$ *Update Forces*
$k(v) \leftarrow \frac{1}{1+k_v v^2}$
$\vec{\mathbf{F}}_{veh}(t) \leftarrow \vec{\mathbf{F}}_{sh}(t) + k(v)\vec{\mathbf{F}}_f(t) + (1 - k(v)\vec{\mathbf{F}}_{sp}(t)$
$\vec{\mathbf{F}}_{tot}(t) \leftarrow \vec{\mathbf{F}}_{nav}(t) + \vec{\mathbf{F}}_{veh}(t)$
Determine acceleration:
$\vec{\boldsymbol{a}}(t) \leftarrow \vec{\mathbf{F}}_{tot}(t)/m$
if $\|\vec{\boldsymbol{a}}(t)\| > a_{max}$:
$\vec{\boldsymbol{a}}(t) \leftarrow a_{max} \cdot \vec{\boldsymbol{a}}(t)/\|\vec{\boldsymbol{a}}(t)\|$
$\vec{\boldsymbol{v}}(t) \leftarrow \vec{\boldsymbol{v}}(t-1) + \vec{\boldsymbol{a}}(t) \cdot T_s$
$\vec{\boldsymbol{v}}(t) \leftarrow v_{max} \cdot \vec{\boldsymbol{v}}(t)/\|\vec{\boldsymbol{v}}(t)\|$

Table 4.1: Parameter Set.

Type	Parameter name	Values
Motivation	$\alpha, v_d, t_r, \boldsymbol{\psi}, \theta_f, \beta$	$(0.8, 2.0, 0.05, [3.0, -0.3], 0.3, 2.2)$
Navigation	k_d, σ_d	$(200, 0.09)$
Shape	M_s, d_{0s}, σ_s	$(800, 4.0, 0.1)$
Flow	M_s, d_{0f}, σ_f	$(600, 6.0, 0.1)$
Speed	$A, \Delta T, \sigma_y$	$(400, 1.0, 0.2L)$
Constraints	a_{max}, v_{max}, m, k_v	$(3.0, 4.0, 75, 0.1)$

4.2 Simulated Experiments

We conduct a series of experiments to validate our methodology. In the first experiment, the SVO effects are analysed in a simplified scenario consisting of one vehicle and one pedestrian. The pedestrian model is taken from [212]. In this way, we can focus on analysing the effects of SVO only, rather than immediately delving into the pedestrian model design. The results of this experiment have been published in [210]. The second set of experiments are concerned with the development of a pedestrian model for AV. We also repeat the procedure of the first experiment for SVO evaluation. This allows us to check if the learnt AV policy is capable of handling the added pedestrian complexity. We also compare two different DRL algorithms on this task.

4.2.1 Experiment 1

A 2-D driving simulator simulator was developed using the Python programming language to validate the proposed method. The simulator was used to train and subsequently test our RL agent. The overall system architecture is represented in Figure 4.1. The simulator models the physics of the problem, i.e. it performs time integration and simulates the interaction between the ego-vehicle and the pedestrian. The simulator is wrapped in an OpenAI Gym [221] environment, which communicates with a SAC agent of the Stable-Baselines3 package [222]. OpenAI Gym Environment is a widely used interface for RL environments. The interface is part of the OpenAI Gym package, which is an open-source package for developing reinforcement learning environments and testing DRL algorithms. Stable Baselines3 [223] is an open-source python package that implements state-of-the-art Reinforcement Learning algorithms.

The driving simulator implements a bicycle model for the vehicle and we model the pedestrian behaviour with algorithm 1. We used a machine with one NVidia GeForce GTX 1080 Ti and a Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz processor to perform the Neural Network Training.

Scenario

The road scenario is represented in Figure 4.2. It consists of a single vehicle and a single pedestrian on a straight road. At the start of each episode, the pedestrian randomly spawns on either the bottom or the top pavement with equal probability and also has a fixed probability of crossing the road. The pedestrian spawn position on the pavement is chosen according to a normal distribution. If the pedestrian crosses, a random goal position on the opposite side of the road is generated according to a normal distribution, otherwise the pedestrian simply walks along the pavement. This way we are able to generate episodes with both crossing behaviours and the RL agent learns to distinguish between them and exploit that to its advantage. We chose a value of 0.9 for the pedestrian crossing probability in order to generate more episodes with actual car-pedestrian interactions, as this kind of interactions are more complex

and the RL agent needs more episodes to learn the correct policy in these situations. The car spawn position and velocity are also chosen randomly according to a normal distribution. The ego vehicle goal position is located along the centre of the lane, 30 m away in front of the ego-vehicle starting point. An episode terminates if the car reaches its goal or if a collision occurs.

Network Training

We trained a total of 9 different policies with different SVO values from 0° to 80°. For each SVO value, an agent is trained for 100,000 steps, at each a tuple consisting of the current observed state, action taken, reward received, and the subsequent next state is stored in a replay buffer. A normally distributed action noise is also added to the actions taken by the agent at training time to favour exploration. The replay buffer size is equal to the number of steps of the entire simulation so that the entire experience gathered by the agent is used during training. The learning rate is initially set to 0.001, then it decreases linearly to 10% of the initial value. The batch size, τ and γ parameters are set to 256, 0.005, and 0.99 respectively.

The neural network architecture consists of two fully connected layers with 256 hidden neurons each, shared by both the actor and critic networks. A simple fully-connected multilayer perceptron network was used, as the input consists of features of the ego-vehicle and the pedestrian rather than raw images. The neural network is trained using the SAC algorithm. The SAC agent performs an action based on the observations provided by the environment and improves the policy at each training step. After executing an action, the environment state is updated and a new set of observations is given to the agent. This cycle repeats until the policy converges.

Results

We carried out computer simulation experiments to evaluate the trained agents. Each agent is tested with 100 training episodes. In order to have comparable results for each agent, the testing episodes share the same initial conditions. Across all the episodes, no collisions with the pedestrians were detected. We compare the trained agents in terms of pedestrian safety and time efficiency in achieving the goal.

Quantitative results

Simply by considering the pedestrian crossing's velocity into the pedestrian reward term, we can see that the car automatically learns a more pro-social behaviour. This more pro-social behaviour corresponds to both increased safety for the pedestrian and an increased time to complete the task, indicating that the car is more likely to slow down and yield to the pedestrian. Figure 4.6a shows the average time taken by the autonomous vehicle to reach their goal. As the SVO increases, the car is more likely to yield to the pedestrian, therefore the average time to reach the goal has an increasing trend. We also computed the minimum distance across all testing episodes

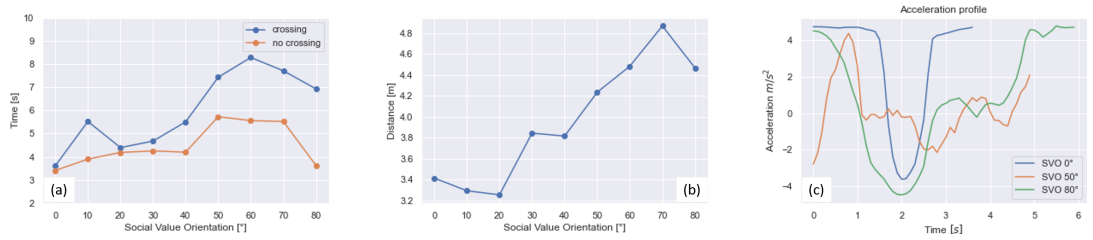


Figure 4.6: (a) Average time to complete the task if the pedestrian is crossing (blue) or if the pedestrian is not crossing (orange). (b) Average minimum distance between the pedestrian and the vehicle. (c) Vehicle acceleration profile with the same episode initial conditions for three SVO values.

for each agent and plotted it in Figure 4.6b. As expected, this parameter also has an increasing trend.

We plot two curves depending on whether the pedestrian is crossing or not in the episodes. For low SVO values, the time to reach the goal is similar for the two curves, because the car arrogantly occupies the lane before the pedestrian is allowed to start crossing. From our simulations, we have seen that an SVO of at least 30° is required to see a significant change in the AV social behaviour, which explains why the two curves are quite similar at the beginning. For higher SVO values, the time to reach the goal when the pedestrian is not crossing also increases. This is due to the fact that the car exhibits a more cautious behaviour and it slows down when close to the pedestrian, even if the pedestrian is not crossing.

Figure 4.6c shows the acceleration profile for an episode with crossing pedestrian and three different SVO values. We can see that overall the acceleration value is lower for the SVO of 80° compared to the one at 0°, indicating that the car moves at a slower speed. Also, for an SVO of 0°, the car actually starts to slow down much later than the SVO of 0°. The acceleration profile for an SVO of 50° oscillates more than the other two. Indeed, the trajectories generated from that agent exhibit a much more hesitant behaviour compared to those at 0° and 80°. In such episodes, the pedestrian also starts hesitating and doesn't commit to any behaviour, which is why the average time to reach goal also decreases for higher SVO values. This reflects a typical scenario in which pedestrian and driver don't come to an agreement and reach an impasse, as they both try to claim the right of passing. This explains why the average time to reach goal decreases again after an SVO value of 60°.

Qualitative results

In Figure 4.7 we visualize the trajectories generated by our simulator in two episodes for three different SVO values. In the first episode the pedestrian is standing on the bottom pavement and is trying to cross the road, whereas in the second episode the situation is reversed. The reader can find a video demo of the trajectories in the supplementary material. We can see that in Figure 4.7a the pedestrian hasn't even

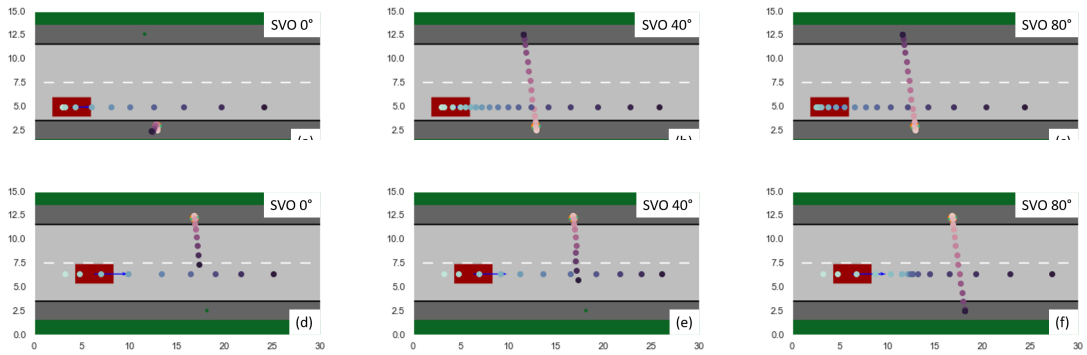


Figure 4.7: Pedestrian and vehicle agent trajectories for two episodes and three SVO values. Figures in the same row refer to the same episode and share the same initial conditions but have different SVO values. The temporal progression is indicated by coloring the trajectories from lighter to darker colors. In Fig. (b), (c), and (f) the car yields to the pedestrian, whereas in (a), (d), and (e) the pedestrian crosses after the car has passed. We can see that the car has a mixed behaviour with an SVO value of 40° (Fig. (b) and (e)).

started to cross the road when the episode is over. This is due to the fact that the low SVO value makes the car behave very aggressively. We can see that in the same Episode but for higher SVO values the pedestrian manages to reach the goal before the end of the episode, meaning that the car yields. The difference between SVO values 40° and 80° is where the car stops yielding to the pedestrian: the car stops almost immediately for an SVO value of 80°, but for an SVO value of 40° the car stops closer to the pedestrian. Similar behaviours can be found in the second episode. We can see that the distance covered by the pedestrian within the end of the episode increases as the SVO value increases and that the pedestrian manages to reach their goal completely in the third figure.

4.2.2 Experiment 2

We divide the experimental results section for this second set of experiments as follows: Sections 4.2.2 and 4.2.2 present qualitative and quantitative evaluations of our pedestrian model. Section 4.2.3 introduces the reinforcement learning scenarios, Section 4.2.3 gives details about the DRL training, and Section 4.2.3 presents the evaluation of the trained agent in the interactive environment.

Gap-Acceptance Validation

We use two real-world pedestrian datasets [84], [104] to evaluate our gap-acceptance model based on motivation. Lee et al. [84] gathered data as part of a virtual reality experiment to investigate how the combination of kinematic information from a

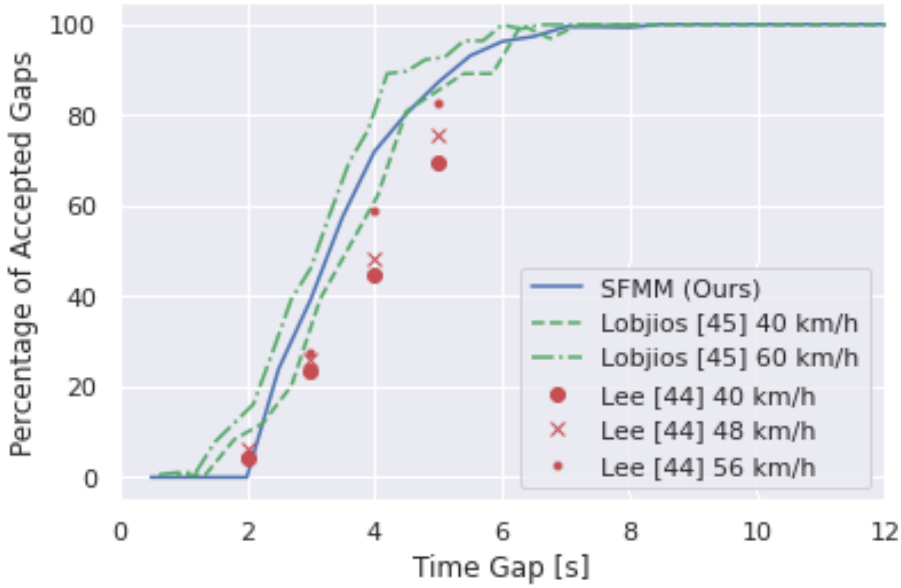


Figure 4.8: Road-crossing probability in Lee et al.'s data [84] (red) and Lobjois et al.'s data [104] (green), together with our model's (blue).

vehicle (e.g., Speed and Deceleration), and eHMI designs, play a role in assisting the crossing decision of pedestrians. The authors of [104] designed a gap acceptance task to investigate the relationship between age difference and accepted gaps. Since age differences is not in focus in this work, we only used the data from the age group 20-30, similar to the age range of participants in [84]. In Fig. 4.8, we show the gap acceptance curve generated by our Social Force Motivation model (SFMM) is in line with the empirical data and is overall capable of capturing both of this datasets well.

Qualitative pedestrian motion analysis

We test the pedestrian behavioural model in the following scenarios:

- fixed AV position lateral interaction;
- fixed AV position frontal interaction;
- slow-speed AV (1-5 m/s) lateral interaction;
- medium-speed AV (10-15 m/s), with three different acceleration values with lateral interaction;

We focused more on the lateral interactions between the vehicle and the pedestrian as we are mostly interested in pedestrian crossing behaviour. For each of the above scenario classes, we performed an evaluation with the pedestrian crossing from both

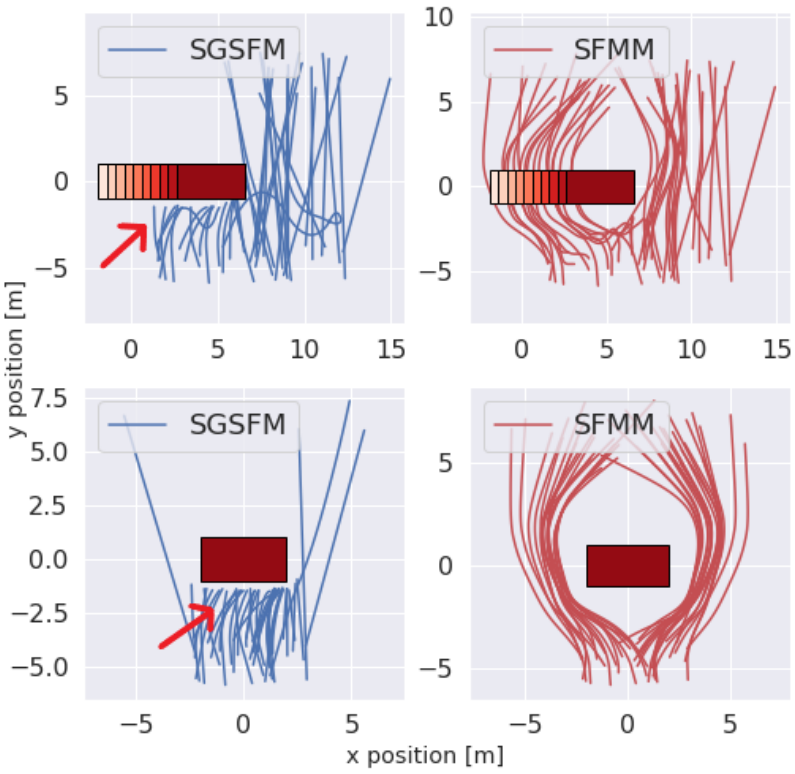


Figure 4.9: Qualitative trajectory comparison between our model (red) and the model in [212] (red). We can see how our pedestrian model is capable of overcoming a static car obstacle whereas the Sub-Goal Social Force Model (SGSFM) gets stuck on opposite side of the car with respect to its goal (as indicated by the arrows). The pedestrian is trying to cross from bottom (negative y values) to the top. The color map from lighter to darker defines the passing of time (light is more in the past).

road sides of the road. In Fig. 4.9 we report qualitative comparative analysis of the trajectories generated by our model (red) and a state-of-the-art social force model [212] (blue). Trajectories are obtained by changing the pedestrian spawn and goal positions, while keeping the same initial conditions for the AV. Our simulations show that the introduction of the $\vec{\mathbf{F}}_{flow}$ term allows the pedestrian to overcome situations in which the repulsive force $\vec{\mathbf{F}}_{shape}$ cancels out the navigational force $\vec{\mathbf{F}}_{nav}$, allowing the pedestrian to overcome a static obstacle. This feature was not present in the previous work [224]. Fig. 4.10 shows additional qualitative trajectories pedestrian trajectories obtained in a frontal interaction (Fig. 4.10(a)) and with car medium-speed (Fig. 4.10(b)). Additionally, we perform a computational analysis of the pedestrian model. The results show that our model computes the pedestrian acceleration in 0.36 +- 0.3 ms against 60 +- 2 ms for the SGSFM model [212]. Good computational performances enable faster DRL training.

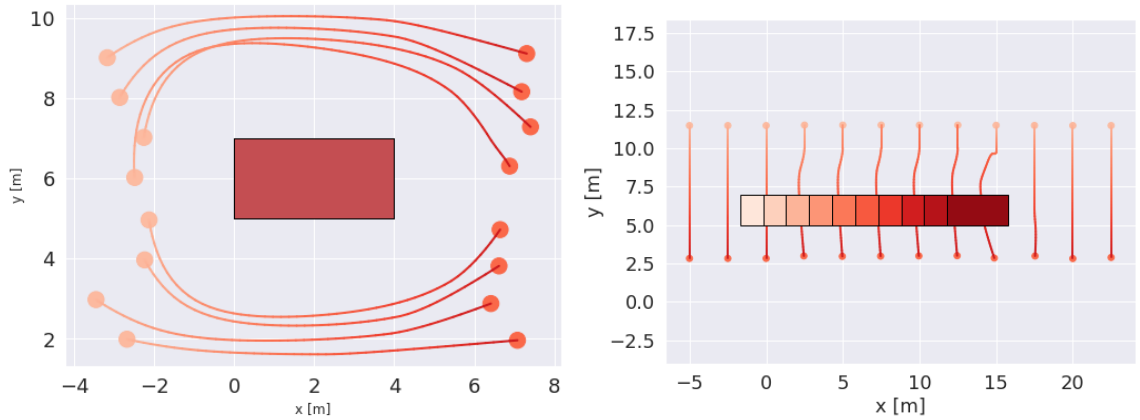


Figure 4.10: Simulation trajectories. The color map from lighter to darker defines the passing of time (light is more in the past). (a) Fixed AV frontal interaction crossing from bottom to top, (b) fixed AV crossing from top to bottom, (c) lateral interaction, (d) slow-moving AV. For each figure, a darker colour indicates a later simulation time. The initial position and goal positions are represented by an orange and a purple circle respectively.

4.2.3 Reinforcement Learning Scenario

We trained and subsequently tested our DRL agent on a straight road scenario with a single pedestrian, (see Fig. 4.2).

We modelled a straight road with the ego-vehicle and a pedestrian. We chose a road length of 60 m and width of 6 m, which is the average road width for a two lane urban road in the UK. The pedestrian can spawn either on the top pavement or on the bottom pavement, whereas the AV always spawns in the bottom lane. This choice does not constitute any loss of generality as we formulate the decision-making

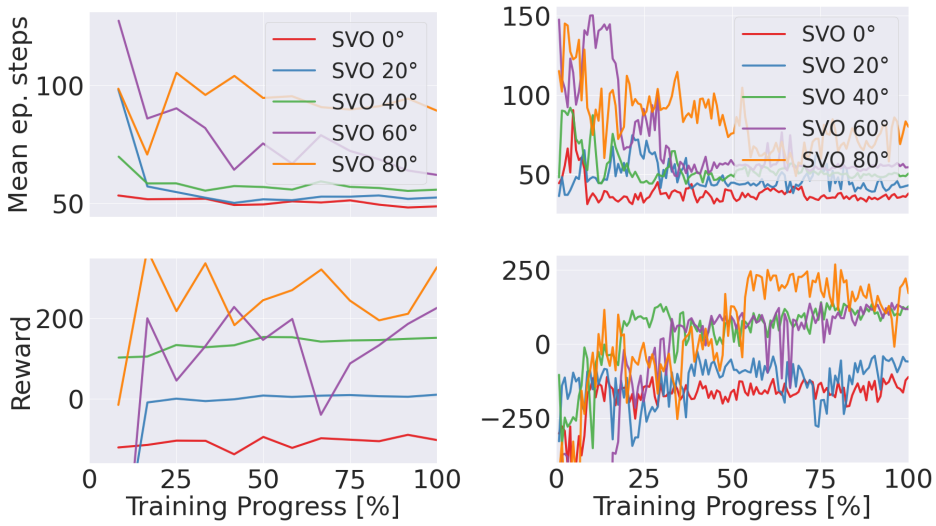


Figure 4.11: Mean episode length in timesteps (top) and mean reward (bottom) with SAC algorithm (left) and PPO (right).

problem in the ego-vehicle reference frame. The AV's initial speed is chosen with a uniform distribution in the interval (0 m/s, 15 m/s) as we are interested in studying a low-speed urban scenario. Selecting random values for the initial conditions favours exploration in the early stages of the training. The pedestrian's initial position along the pavement is sampled from a uniform distribution. The pedestrian's goal position is always on the opposite side of the road from their spawn point and is sampled from a normal distribution with mean value equal to the pedestrian position to ensure the distance from the crossing point is not excessive.

Neural Network Training

The neural network architecture is the same for both PPO and SAC and consists of two fully connected layers with 256 hidden neurons each, shared by both the actor and critic networks. A simple fully-connected multi-layer perceptron network was used, as the input space is simple enough to allow us to use a simpler neural network rather than a Convolutional Neural Network which would be harder to train.

We observed that by directly training the RL agent with the new pedestrian model resulted in only aggressive policies for the RL agent, even for SVO values close to 90°. The DRL algorithm used to get stuck in a local minimum, which caused limited exploration: since the newly introduced pedestrian model has a more cautious behaviour compared to our previous work [224], the AV agent optimised only the first term of eq. 4.3, neglecting the pedestrian's reward. We solved this issue by splitting the training in two parts. For the half of the training, the model is trained with a reckless pedestrian model that always crosses the road. For the second half of the training, we switch the pedestrian model to the more complex one. The idea

behind this is that the RL model is more cautious when the conservative pedestrian is introduced, which allows to explore braking actions without falling into a local maximum for the reward. In this way, agents with higher SVO values learn that breaking yields to higher altruistic rewards.

We compare the performances of two different RL algorithmsby training a total of 10 different policies: 5 for the SAC algorithm and 5 for the PPO algorithm with SVO values of 0°, 20°, 40°, 60°, and 80° respectively. In general, the SAC algorithm requires longer than PPO to train a policy that yields the same cumulative reward, but requires fewer steps. We compared the two algorithms by keeping the total computation time constant. We trained each policy for roughly 150 minutes, which resulted in a total of 2.5×10^6 steps for PPO and 2.5×10^5 steps for SAC. A normally distributed action noise is also added to the actions taken by the agent during training time to favour exploration. We set the replay buffer size for the SAC algorithm equal to the number of training steps so that the entire experience gathered by the agent is used during training. We choose a linear decay for the learning rate, initially set to 3×10^{-4}. The discount factor γ was set to 0.99.

We show the training curves for both PPO and SAC in Fig. 4.11. The figures show the mean episode length and the total reward gathered for different SVO values. We observe how at the end of the training the two algorithms return comparable results both in terms of mean episode length and reward gathered, which shows consistency between training instances. However, we note that for some of the SAC policies, the reward is not entirely stable at the end of the training, indicating that the SAC algorithm is much more time consuming than PPO. Nonetheless, the SAC policies yield acceptable results in terms of policy behaviour, allowing for comparison of the two algorithms. Figure 4.12 shows the acceleration profile for an episode with crossing pedestrian and three different SVO values. We can see that overall the acceleration value is lower for the SVO of 80° compared to the one at 0°, indicating that the car moves at a slower speed. Also, for an SVO of 0°, the car actually starts to slow down much later than the SVO of 0°. The acceleration profile for an SVO of 50° oscillates more than the other two. Indeed, the trajectories generated from that agent exhibit a much more hesitant behaviour compared to those at 0° and 80°. In such episodes, the pedestrian also starts hesitating and doesn't commit to any behaviour, which is why the average time to reach goal also decreases for higher SVO values. This reflects a typical scenario in which pedestrian and driver don't come to an agreement and reach an impasse, as they both try to claim the right of passing. This explains why the average time to reach goal decreases again after an SVO value of 60°.

Mutual Interaction Evaluation

We create two test suites of 1000 testing episodes to evaluate the effect that our SVO reward design has on the agent behaviour. In half of the episodes, the pedestrian crosses the road from top to bottom and in the other half from bottom to top. The first one is used to evaluate the agent with our pedestrian model. In the second one,

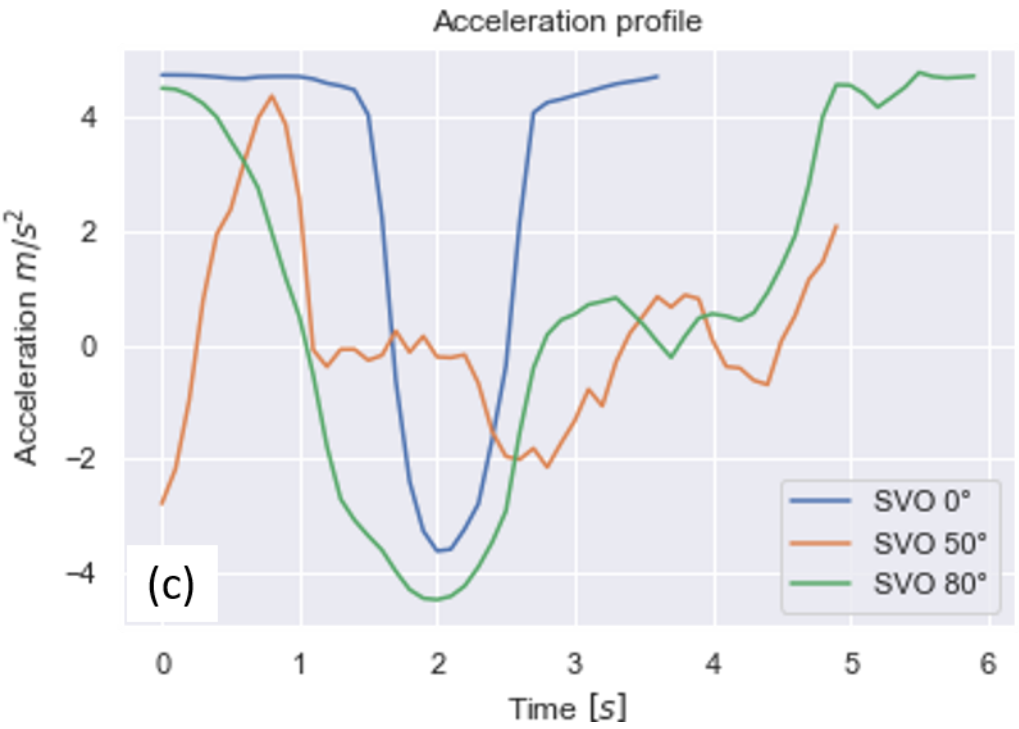


Figure 4.12: Acceleration profile with different SVO values.

we increase scenario complexity by making the pedestrian unaware of the vehicle's presence, i.e. crossing regardless of the vehicle's position and speed. In this way, we were able to include hazardous and unexpected scenarios that will stress the controller robustness to the pedestrian model. We analyse the smoothness of the agent trajectory, how its behaviour is affected by SVO, and the agent's robustness to the pedestrian model. Agents with an SVO value of 0° serve as a baseline for State of the Art DRL methods with traditional reward functions that only take the ego-vehicle's goal into account, as an agent with an SVO value of 0° is exactly equivalent to a standard DRL agent.

Results

Qualitative results

In Fig. 4.13 we show pedestrians and AV trajectories with different SVO values with the same initial conditions. Agents trained with SAC (first row) and PPO (second row) display similar trajectories. In Fig. 4.13(a) and (d) the vehicle accelerates to prevent the pedestrian from crossing due to a low SVO value. Viceversa, in Fig. 4.13(c) and (f) the ego-vehicle displays a behaviour called early-stopping, in which it slows down to let the pedestrian initiate crossing. Fig. 4.13(b) and (e) have

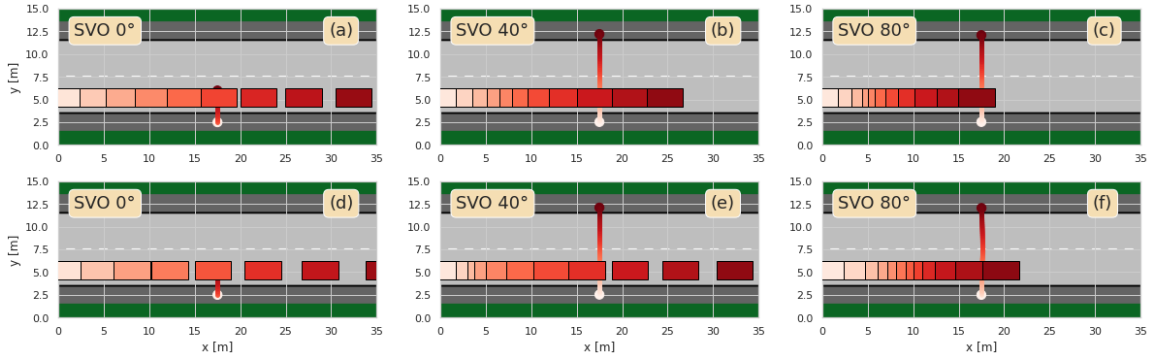


Figure 4.13: Pedestrian and vehicle agent trajectories for two episodes and three SVO values. Fig. (a)-(c) are generated with SAC and (d)-(f) with PPO. The temporal progression is indicated by coloring the car and pedestrian's trajectories from lighter to darker colors. In Fig. (b), (c), (e) and (f) the AV yields to the pedestrian, whereas in (a), (d) the pedestrian crosses after the AV has passed and has not completed crossing when the episode terminates. We can see that the 80° SVO has a less aggressive behaviour than 0° and 40°.

intermediate behaviour. The effects of the SVO with the overall agent behaviour are in line with our expectations, i.e. pro-social behaviour for high SVO values and egoistic behaviour with low SVO.

Fig. 4.14 shows the qualitative effect that unpredictable pedestrian behaviour has on an agent with SVO 0°. Fig. 4.14(a) and (c) have an aware pedestrian, Fig. 4.14(b) and (d) an unaware pedestrian. Despite the fact that the controller SVO is 0°, the car stops to let the pedestrian cross in order to avoid collision, thereby favouring safety over its own egoistic behaviour.

Quantitative results

First of all, we evaluate the agents success rate in completing its task in the first and second test suites. We consider an episode successful when the ego-vehicle reaches the end of the road whilst avoiding the pedestrian. All the agents successfully completed the task without collisions with the pedestrian in both the first and second test suite, which demonstrates the fact that our model is capable of handling the added complexity of risky scenarios.

In principle, two RL algorithms that solve an MDP problem should both yield optimal policies which achieve the same cumulative reward Fig. 4.11. However, actions taken are not necessarily the same. Agents trained with PPO showed smoother acceleration profiles, as shown in Fig. 4.12, consistently with DRL theory. For an AV passenger, the policies generated by PPO seem to be more comfortable from an ergonomics perspective, a fact that we intend to investigate in future research. However, SAC has better exploration strategies, rendering it more suitable to solve complex tasks.

Fig. 4.6 shows the average minimum distance between the ego-vehicle and the

pedestrian. The distance increases as the SVO increases, which indicates that the AV has a more altruistic behaviour and yields to the pedestrian. We observed that the policies trained with the SAC algorithm tend to stop much earlier to yield to the pedestrian compared to the PPO algorithm, offering an explanation to why the average minimum distance are significantly larger for such policies.

Overall, the results are consistent with our previous findings [224], which confirms that the agents are capable of learning behavioural strategies with more complex pedestrian behaviour while still being able to handle risky or unexpected scenarios, which is promising for real world applications.

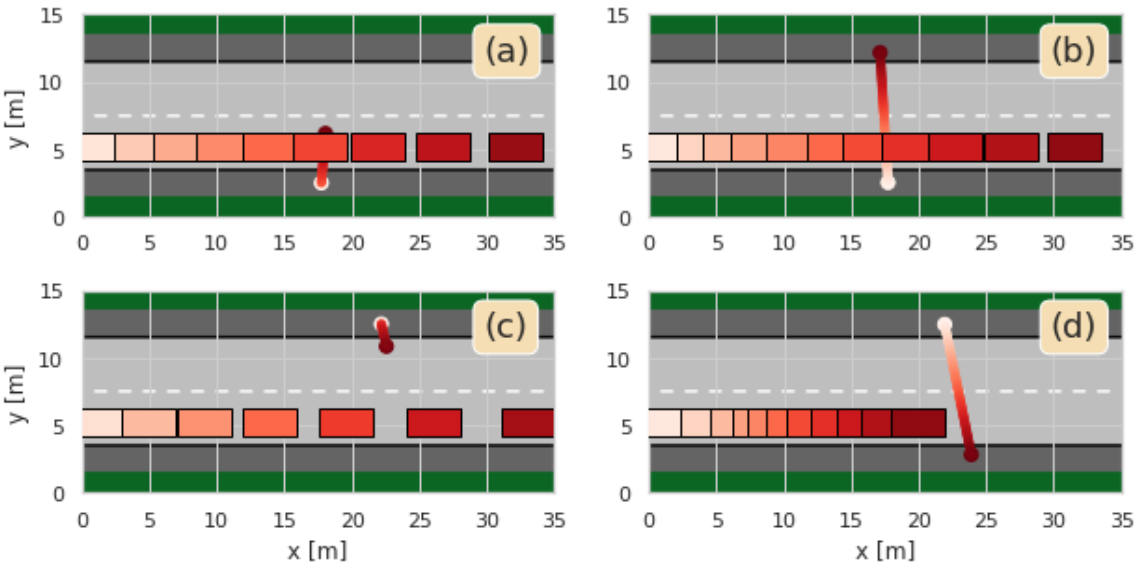


Figure 4.14: Qualitative trajectories with unaware pedestrian (b), (d) and aware pedestrian (a), (c). Figures on the same row share the same initial conditions. The ego-vehicle agent is the same for all scenarios (SVO 0°) and is capable of distinguishing exploitable pedestrian behaviours from hazardous ones.

4.3 Conclusions

We presented an approach to solving the pedestrian collision avoidance problem in a scenario consisting of a vehicle and a single pedestrian using a state-of-the-art Reinforcement Learning algorithm called Soft-Actor Critic. We demonstrated that by including Social Value Orientation in the RL reward function design, the trained vehicle agent naturally displays human-like behaviour, such as yielding and early stopping. The SVO value affects how much the trained agent is likely to yield to the pedestrian. By blending individualistic and altruistic components into the reward function design, our approach ensures that the AV is not merely a self-serving entity but a responsible and socially aware road participant, making decisions that mimic the

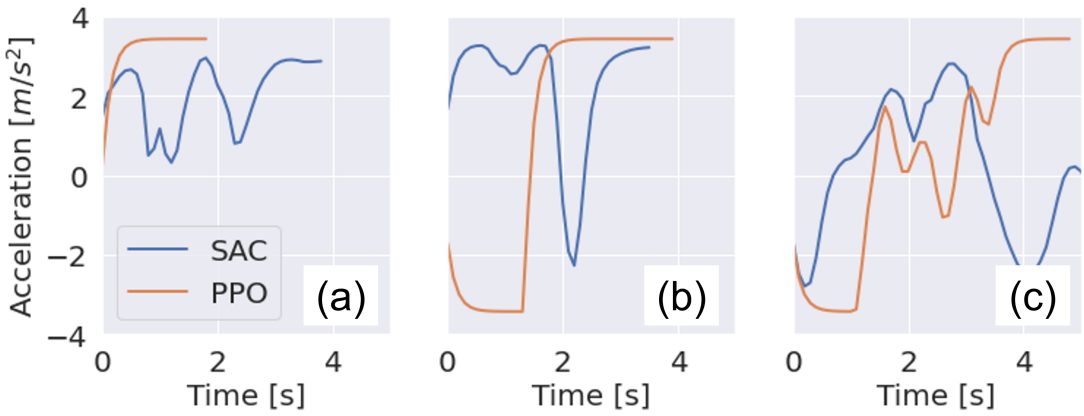


Figure 4.15: Acceleration profiles for PPO and SAC policies on the same testing episode with SVO values of 0°-(a), 40°-(b), and 80°-(c).

thoughtfulness and adaptability exhibited by human drivers. This innovative fusion of technology and ethical considerations is a promising step toward the development of AVs that seamlessly integrate into our complex and dynamic road environments.

We introduced a novel pedestrian model for computer simulation that joins gap-acceptance and social-force models that incorporates a situational awareness risk evaluation to initiate crossing. We demonstrated how the DRL agent is still capable of handling more complex human models, which is an important prerequisite in order to handle real pedestrians. We have also conducted a comparative analysis of two different model-free DRL algorithms (SAC and PPO) designed for continuous actions spaces applied to our problem.

We have shown how PPO policies lead to smoother actions which are more appealing from an ergonomics perspective and offer improvements with respect to previous papers that applied DRL to our problem [12], [169]. This work also highlights how SVO can be an effective tool to design DRL algorithms in human-machine interaction applications. A limitation of our current work is that the SVO policies are trained with discrete SVO values and one would have to switch controllers to alter the ego-vehicle behaviour. We further intend to investigate whether SVO can be used as an input parameter for the neural network, rather than being a fixed parameter at the beginning of each training. This would allow for continuous changes in the car behaviour and the usage of a single controller architecture.

The main assumption in this work is the presence of a single pedestrian. An immediate extension of this work will be to tackle the presence of multiple pedestrians and vehicles, which will be done in Chapter 5 of this Thesis. Further, we are looking to improve the state-space-representation and utilise more advanced neural network architectures to validate our model.

Chapter 5

Game-Theoretic Strategies for AV Decision-Making in Multi-Agent Scenarios

Chapter 4 introduces a RL framework for training tactical decision-making agents. As discussed in Chapter 1, an important advantage of learning based methods is that they can scale to different driving scenarios. One of the main limitations assumed in Chapter 4 was the fact that it focused on a single agent problem. In this Chapter, I extend the SVO formulation introduced in Chapter 4 to multi-agent settings and its effect on traffic flow and performance metrics is analysed. In order to do so, level-k game theory (see 2.6) is employed to jointly train a pedestrian and a vehicle agents. To the best of our knowledge, this is the first time this has been done in the literature.

Indeed one of the main limitations in multi-agent scenarios involving pedestrians is the lack in the literature of a universal pedestrian model that would describe their behaviour in multi-lane settings that include numerous vehicles [225]. In this Chapter, a pedestrian model is trained via reinforcement learning to overcome this issue. It is important to stress that the focus of this Chapter is not the development of a universal pedestrian model. Instead the pedestrian model serves as part of an episodic driving simulator for the development of AV decision-making algorithms based on DRL. Therefore, further studies will need to focus on the development of realistic pedestrian models for DRL simulators. Here the pedestrian model is used to train the ego-vehicle in scenarios with multiple agents surrounding it.

In the previous Chapter, the MDP state space only consisted of vehicle and pedestrian coordinates in a vector ($s_t \in \mathbb{R}^n$). In this Chapter however, since the number of surrounding vehicles can dynamically change, a graph is constructed to represent the environment around the ego-vehicle and processed via a GNN. Ablation studies on the network architecture are carried out and its performance is evaluated against other type of NN architectures (MLP and CNN). A neural network architecture that takes surrounding agents into account as well as ego-trajectory and goal destination into account is proposed.

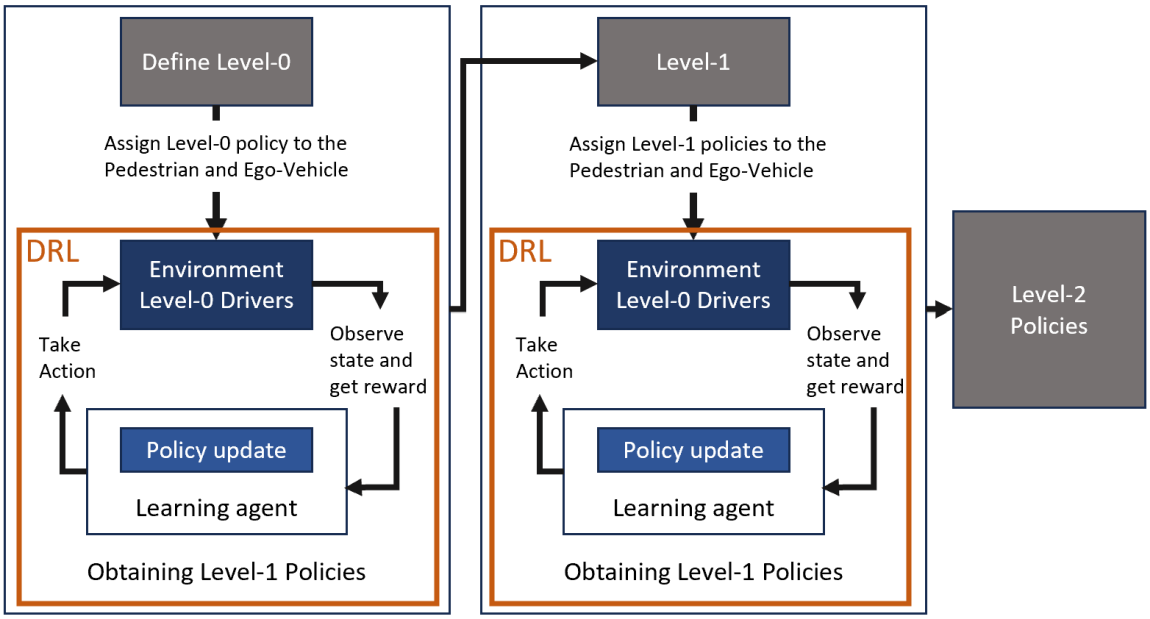


Figure 5.1: Multi-agent framework to train ego-vehicle and pedestrian policy networks. The DRL framework is used to obtain policy with different levels, according to the Level-k game theory. The policies are Graph Neural Networks.

This Chapter is structured as follows. Section 5.1 illustrates the methodology developed in this study. In Section 5.2 we describe the experiments that were carried out. Finally, in Section 5.3 we draw some conclusions, highlighting the research findings, current challenges and future directions.

5.1 Methodology

This Section will describe the methodology developed. The framework, consisting of DRL, Level-k game theory is elucidated in 5.1. Firstly, we will introduce the multi-agent setting problem. Secondly, we describe how Graph Neural Networks can be used to model the environment surrounding the ego-vehicle and we introduce our novel network architecture. Then we describe how the SVO reward function introduced in Chapter 4 can be extended to multi-agent scenarios. Finally, we illustrate the learning procedure that combines DRL and Level-k Game Theory.

5.1.1 Problem Description

The selection of the unsignalized intersection depicted in Figure 5.2 as the focal scenario for this study stems from a deliberate choice, driven by the inherent complexity it presents when compared to other traffic scenarios typically characterized by the presence of traffic control devices, such as traffic lights or road signs. This specific

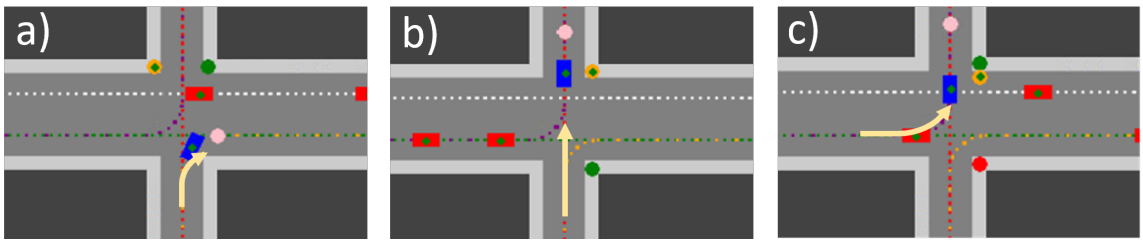


Figure 5.2: The training and test scenarios that are used to evaluate the GNN agent. Scenarios a) and b) are used for both training and testing, whereas scenario c) is only used to test the generalisation of the GNN agent. The pink circle corresponds to the goal position of the agent. The orange circle represents a pedestrian. The starting position and goal position of the pedestrian are indicated with a green and red circle respectively.

intersection scenario encapsulates a dynamic and intricate web of vehicular and pedestrian interactions that demand a nuanced and adaptive approach to navigation. The scenario simulator is developed in Python and is based on the 2D driving simulator CARLO [27].

In this scenario, both the ego-vehicle and the pedestrian exercise their discretion in determining the opportune moment to enter the intersection area. It is worth noting that this aspect elevates the complexity of the scenario, as it introduces a dynamic element into the decision-making process for both the autonomous vehicle and the pedestrian. In essence, the chosen scenario serves as a testing ground for the development and evaluation of autonomous driving systems, as its inherent complexity ensures that the research conducted within its context contributes meaningfully to the field of autonomous driving. Insights and solutions originating from this study can be extended beyond the confines of this specific intersection and into the realm of autonomous navigation in diverse and intricate urban environments.

The scenarios consists of 5 main paths for the vehicles and 4 main paths for the pedestrian. The blue car represents the ego-vehicle while the red cars follow an Intelligent Driver Model (IDM). The IDM is a time-continuous car-following model for the urban traffic simulation [226]. The pedestrian is represented with an orange circle. The pink circle represents the ego-vehicle goal, whereas the pedestrian starting and goal positions are represented by a red and green circle respectively (see Figure 5.2c).

Non-ego vehicles can travel along the main road in two opposite directions (two-lane road). The ego-vehicle has three main tasks to complete: merging Figure 5.2a, going straight across the intersection b), and performing a left lane turn. The tasks are hard as they require the vehicle to negotiate with upcoming traffic as well as with the pedestrian. The pedestrian task consist in moving from their starting position to their goal position. The pedestrian can spawn on any corner of the intersection and cross the intersection along each of the four main road but not diagonally. The

red vehicles corresponding to non-ego vehicle occupy the main road. The number of vehicles on the road and the initial conditions (position and velocity) of all agents is chosen randomly at the start of each episode to increase variety in the episodic samples and introduce an additional challenging factor.

5.1.2 Graph Model of Traffic

As mentioned in Section 1.3 it will be assumed that the detection and scene understanding from raw sensor data are carried out by a perception module. As such, the information regarding surrounding agents positions, velocities, and class (vehicle or pedestrian) is assumed to be available to the ego-agent.

The interactions among vehicles in the intersection are denoted by an undirected graph. Supposing that there are N agents in the scene, we define a spatial graph $G_t = (V_t, E_t)$, where $V_t = \{v_t^n | \forall n \in \{1, ..., N\}\}$. Each vertex represents an individual agent in the scene with its associated features. Node feature selection in Graph Neural Networks (GNNs [227]) refers to the process of choosing a subset of relevant features for each node in a graph before applying a GNN model. Node features considered in this study include: agent's class $c_i \in \{\text{pedestrian, vehicle}\}$, agent's position (x_t^n, y_t^n), velocity $(v_{x,t}^n, v_{y,t}^n)$, orientation θ_t^n, and distance from the closest agent $d^n = \arg\min_{j \in 1,...,N} ||\boldsymbol{p}_n - \boldsymbol{p}_j||$, where $\boldsymbol{p}_i$ indicates the position of agent i. E_t indicates the set of all edges, which represents the mutual interactions between vehicles. Ablation studies have been conducted to find out which subset of this node features are most suitable for learning. Each agents i node features are therefore:

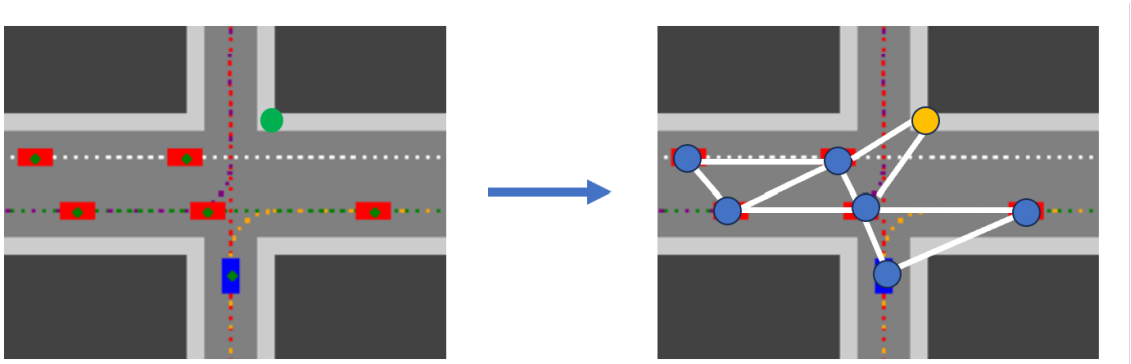


Figure 5.3: A graph structure is built from the traffic scene that includes all agents within vehicle detection radius.

$$\boldsymbol{x}^n = \left[c_i, x_t^n, y_t^n, v_{x,t}^n, v_{y,t}^n, \theta_t^n, d^n\right] \tag{5.1}$$

Figure 5.3 shows an example of the constructed graph. Blue nodes represent vehicles whereas orange nodes pedestrians. Only agents within ego-vehicle perception distance are included in the graph, which we assumed to be 100 m. LiDAR detection

distances can typically reach 150 m in AVs, therefore 100 m is a distance that covers most modern LiDAR systems for AVs. Edges are used to capture mutual influences between nodes. Therefore, an edge between two nodes is created if their relative distance is less than an influence radius, which is a tunable hyperparameter. We have empirically found out that influence radii between 30 m and 100 m improve the quality of the agent's performance (see Section 5.2.2).

5.1.3 Network Architecture

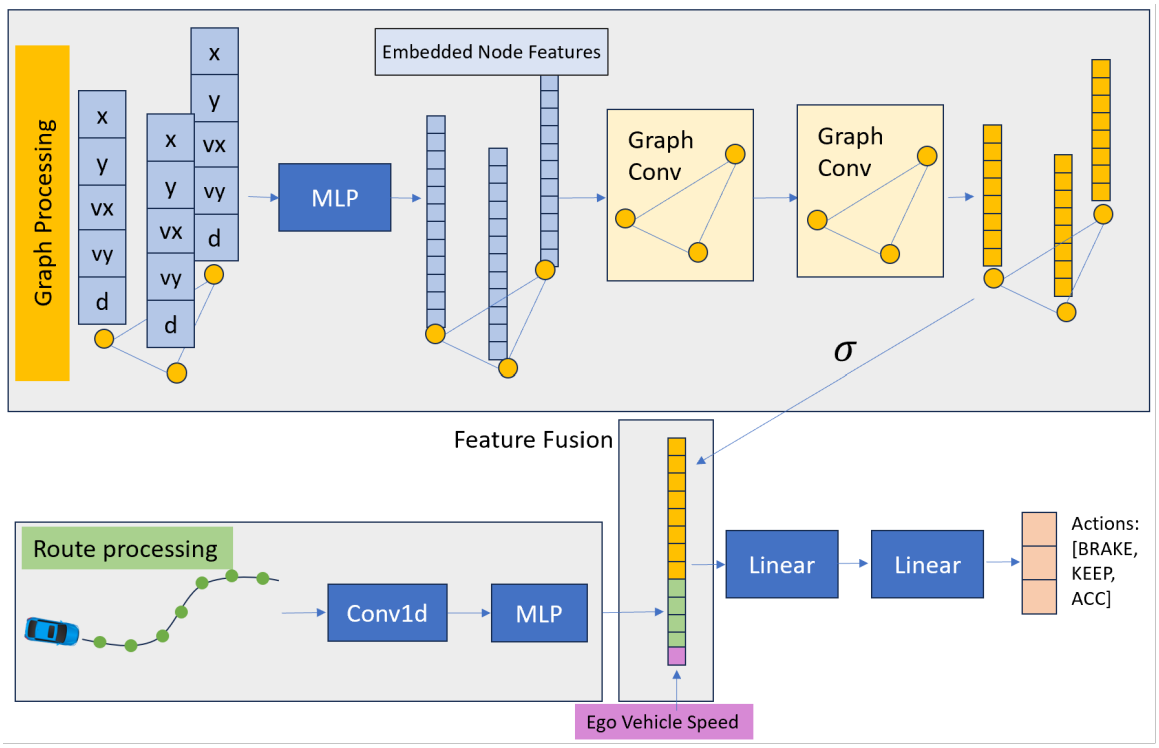


Figure 5.4: The proposed interaction-aware neural network architecture used for DRL policy.

The proposed network architecture is shown in Figure 5.4. The network maps sensory inputs to ego-vehicle actions and is trained via Deep Reinforcement Learning with the DDQN algorithm. The action space is discrete and consists of three possible actions: brake, accelerate, and maintain current speed. The sensory input consists of traffic information (graph input described in Section 5.1.2, route information, and ego-vehicle current speed.

The graph processing is as follows. First the node features are embedded via an MLP (MultiLayer Perceptron) layer. Graph convolutional layers process the embedded node features without altering the graph structure of the data. Node features are then fused together with a single aggregation layer σ.

The route information consists of a set of waypoints describing the path that the ego-vehicle should follow. In robotics path planning, a waypoint is a specific intermediate location or point in space that a robot needs to pass through or reach as part of its overall trajectory or path. Waypoints play a crucial role in guiding the robot from its starting point to its destination while avoiding obstacles and optimizing its movement. The route information is processed by a Conv1d Layer and by an MLP layer.

Route information and the traffic graph are firstly processed and then joined together by concatenation (feature fusion). The ego-vehicle speed is also included in this feature fusion vector. The joint feature vector is processed by two linear layers which output the actions $Q(s, a)$ scores.

5.1.4 Reward Function for Social Interactions

As previously described in Chapter 4, the designed reward function will be split into two main terms:

$$r(s_t, a_t) = \cos\varphi \cdot r_{self}(s_t, a_t) + \sin\varphi \cdot r_{others}(s_t, a_t) \tag{5.2}$$

where r_{self} is the selfish reward term, whereas r_{others} is used to model the utility that the agent attributes to surrounding agents. Note that since the pedestrian model is also being trained in this study, the same reward structure is valid for both pedestrian and vehicle agents.

The first term r_{self} in the reward function is also a combination of multiple terms:

$$r_{self}(s_t, a_t) = r_c + r_g + \alpha_v r_v + r_t \tag{5.3}$$

where r_c is a penalty in case of collision, r_g is the reward for reaching the goal, r_v is a speed reward that encourages the AV to complete the task as quickly as possible, r_t is a timeout penalty, finally α_v is a weight factor for the velocity term. Since the pedestrian and the ego-vehicle will have very different speeds we keep the same reward structure for r_v but use different weights α_v. The timeout penalty is added in the multi-agent scenario to prevent the vehicle from standing still to collect reward by letting other agents pass, which would not occur in the two agent scenarios since once the pedestrian finished crossing there were no more agents to collect reward from. More details on the reward parameters are given in Section 5.2.3.

The second term of Equation 5.2 is used to capture other agent's intentions and comfort in the decision-making process. We extend the previous social-reward term in Chapter 4 to include multiple agents. If there are N agents in the scene (excluding the ego-agent) and indicate the social reward for agent i as r_i, the proposed social reward functions assumes the form:

$$r_{others} = \sum_{i=1}^{N} \frac{1}{d_i} r_i \tag{5.4}$$

d_i is the distance of agent i from the ego-agent. Only agents belonging to the observation graph $\mathcal{G}$ are considered for this weighted sum. The weighting factor $1/d_i$ is added to the sum so that agents that are far away from the vehicle are taken into less account in the decision-making process. This is done to reflect the fact that the ego-agent has less agency on agents that are further away from it.

The assumption is that r_i will be the same as the ego-agent's selfish reward function. Similarly to the ego-agent, other agents will also try to avoid collisions and travel at their desired speed. However, since the desired speed of other agents and their goal are unknown to the ego-vehicle, the term r_i is simplified to have two main terms: a collision penalty r_c and a speed reward term proportional to the agent's speed r_v

$$r_i = r_c + r_v \tag{5.5}$$

.

5.1.5 Decision-Making based on Game Theory and DRL

Level-k game theory [171], or cognitive hierarchy theory, models how people make decisions in strategic situations. It starts with level-0 thinkers who act without considering others, followed by level-1 thinkers who assume opponents are level-0. This pattern continues to higher levels, often up to level-3 in human experiments. This approach approximates Nash Equilibrium and explains deviations when rationality varies. It offers insights into real-world decision-making beyond strict Nash equilibrium assumptions.

To create agents with different decision-making abilities, the DDQN algorithm is used in the simulator. In this setup, the ego-agent acts as a level-k learner, while other agents follow trained level-(k-1) policies or a predefined level-0 behaviour. We start with a predefined level-0 policy as the foundation, and then derive other hither-level policies using DDQN method. For instance, to get a level-1 policy, a traffic scenario where all driver except the ego-agent operate at level-0. The ego-agent learns to respond optimally to this level-0 policy.

Once the training is done, the ego-agent becomes a level-1 agent. The process is repeated to obtain level-2 policies (i.e. the other agents in the scene are initialised to level-1 or level-0 policies and the ego-agent is trained). Surrounding agents choose policies from trained models following a uniform distribution. The framework in Figure 5.1 illustrates this procedure.

The hierarchical learning process outlined earlier reduces computational expenses. At each learning stage, agents other than the ego agent employ pre-trained policies, essentially becoming part of the environment. This allows for the creation of traffic scenarios where agents at varying skill levels are all making strategic decisions simultaneously. This stands in stark contrast to traditional decision-making approaches in congested traffic, where one driver assumes the role of a strategic decision-maker, while others simply follow predefined policies that adhere to specific motion rules.

The training algorithm with Level-k Game Theory is highlighted in Algorithm 2. We combine Level-k game theory with Social Value Orientation to study how the latter affects the interactions between the pedestrians and the ego-vehicle, as well as the overall ego-vehicle capability of completing the task. It is to note that when training a Level-k policy, all other agents follow a level-(k-1) policy and the SVO associated to this level-(k-1) policy is sampled uniformly from a population of agents with different SVO values. This ensures that the ego-vehicle learns an optimal policy with respect to all possible pedestrian combinations. For example, when training a level-2 ego-vehicle with a specific SVO value, the training procedure is the following. During training, at the start of each episode, the pedestrian policy is sampled randomly amongst all level-1 pedestrian policies, consisting of level-1 policies with various SVO values. At the end of the training process, an ego-vehicle policy is obtained for a specific SVO value. The process is repeated to obtain ego-vehicle level-2 policies for all SVO values.

Algorithm 2 Obtaining the Level-k Policy for ego-vehicle or pedestrian

Require: Set of SVO values $\mathcal{S}$, Level-$(k-1)$ policies for the opponent (pedestrian or ego-vehicle respectively): π_{k-1}^{φ} with $\varphi \in \mathcal{S}$

output Policy π_k^{φ} for the agent with $\varphi \in \mathcal{S}$ ▷ *Ego-agent SVO φ is fixed throughout the training procedure*
Initialise primary network Q_θ, target network $Q_{\theta'}$, replay buffer $\mathcal{D}$, social reward function $R_\varphi(s,a)$, $\tau \ll 1$;
for episode $= 1, ..., M$ **do**
  Select random SVO for other agents $\varphi' \sim \mathcal{S}$
  for $t = 1, ..., T$ **do** ▷ *T is the termination step*
    Sampling Steps
    with probability ε, select a random action a_t, otherwise select $a_t = \arg\max_{a \in \mathcal{A}} Q_\theta(s_t, a)$
    Execute action a_t for ego-agent and action $a'_t \sim \pi_{k-1}^{\varphi'}$
    Observe reward r_t^{φ} and new state s_{t+1}
    Store transition $(s_t, a_t, r_t^{\varphi}, s_{t+1})$
    Update Steps
    Sample random minibatch $\mathcal{B}$ of transitions $(s_j, a_j, r_j^{\varphi}, s_{j+1}) \in \mathcal{D}$
    for $j \in \mathcal{B}$ **do**
      Set $Q^*(s_j, a_j) = r_j^{\varphi} + \gamma Q_\theta(s_{j+1}, \arg\max_a Q_{\theta'}(j+1, a'))$
    Perform a gradient descent step on $\frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \left(Q^*(s_j, a_j) - Q_\theta(s_j, a_j)\right)^2$
    Update target network parameters:
      $\theta' \leftarrow (1-\tau)\theta' + \tau\theta$
The policy is determined from the primary network:
  $\pi^{\varphi}(s) = \arg\max_a Q_\theta(s, a)$

5.2 Experimental Results

We developed a 2D driving simulator that was used to train and test our DRL agents. The simulator is based on CARLO [27], a 2D lightweight driving simulator, and is developed with Python. Pytorch and Pytorch Geometric were used to implement the DRL algorithm with Graph Neural Networks, since at the time of development no Python DRL packages allow the use of graphs as input for DRL policies.I used a machine with Intel(R) Core(TM) i7-11700 @ 2.50 GHz and an NVIDIA GeForce RTX 3060 Ti GPU for training and testing our methodology.

5.2.1 Experiments description

We divide the experiments in two major groups. The first group of experiments are aimed at determining the network architecture and the subset of node features to be used to achieve the best Graph Neural Network performance, as well as the training hyperparameters that best suits the GNN needs.

For this first group of experiments, the task for the ego-vehicle is the same, however the pedestrian is not present in this task. The reason behind this choice is twofold: firstly, the ego-vehicle must be able to complete the task in the absence of the pedestrian, as real world scenarios can consist of vehicles only; secondly, as previously mentioned, the pedestrian is an additional agent that needs to be trained. Therefore its presence is an additional variable in the experiments evaluation and we are only interested in comparing the network architecture with respect to baselines and conduct ablation studies on the network. In this first group of experiemnts, we conduct ablation studies on the network parameters and compare its performances against other neural network architectures. The network is compared against an MLP and a CNN. In this experiment, the task are the same as described in Section 5.1.1 without the pedestrian involved. Therefore, the ego-vehicle's task is to merge into traffic as in Figure 5.2a and b during training and scenario c at testing. Ablation studies are also conducted on the input features, i.e. graph, route, and ego-vehicle speed features, as well as on the node graph features (see Eq. 5.1.

The second group of experiments aim at studying the effects of SVO (Section 5.1.4) on the interactions between the ego-vehicle and the pedestrian. We use the network architecture that was developed in the first group of experiments to train both the pedestrian and the vehicle with the Level-k game theory framework [171]. We show qualitative trajectories obtain with the method and the effects of SVO on the ego-vehicle performance. The ego-vehicle performances are evaluated in scenarios where the pedestrian policies are extracted from a population of pedestrians with different SVO values. The training procedure is shown in algorithm 2.

The metrics used to compare results from different neural network architectures are the collision rates, the number of completed episodes and the number of timed-out episodes. For the neural network architectures, we compare our GNN based model against an MLP and a CNN policy. The performances are evaluated on a

suite of 1000 episodes. Each episode has randomly allocated initial conditions, but the same test suite is used to assess different neural networks. In this way, the networks can be evaluated on different initial conditions to test their generalisation performances. At the same time, the same test suite ensures that their performances are comparable across different network architectures. The outcome of each episode can either be collision, success, or timeout. The outcome is collision if the ego-vehicle collides with surrounding traffic. In this case, an episode instantly terminates and is counted as a failure. The timeout indicates that the ego-vehicle did not move within an allocated timeout windows, which we set to be 40s for this task. If the ego-vehicle manages to reach its goal within the predetermined time-window and no collision occurs, the episode is counted as **completed**. The metrics are evaluated on both the training and test scenarios. The test scenarios are used as means to evaluate the network generalisation performance to unseen conditions. It is to note that more scenarios need to be introduced in the training and test sets to make the network policy capable of generalising to a bigger variety of driving conditions, such as roundabouts or crossroads with different topology from the one considered in this study.

5.2.2 Experiment 1

Comparison with Baselines

We compare our model architecture (see Fig. 5.4) with a Multilayer Perceptron and with a Convolutional Neural Network. Since the MLP has a fixed input size, we consider the input vector to consist of the features of the four agents closest to the ego-vehicle. The features for this comparative analysis consists of the four closest surrounding agents' positions and velocities stacked in a vector:

$$\boldsymbol{x}_{\text{MLP}} = \left[p_x^1, p_y^1, v_x^1, v_y^1, ..., p_x^4, p_y^4, v_x^4, v_y^4\right]$$

The CNN input consists of an image obtained from the simulator itself. The raw frames are preprocessed by first converting their RGB values to gray-scale and then by down-sampling the image to a size of 84x84 pixels, similarly to [16]. Since a raw image does not contain velocity information regarding surrounding agents, four successive frames are stacked with each other to create an 84x84x4 input representation. The CNN architecture used in [16] is then used to train the agent. The GNN input is described in Section 5.1.2 and constists of a graph representation of the traffic. In order to have comparability between the GNN and the MLP, the node features used consists of agents positions and velocities. We performed hyperparameter tuning on all models with manual search. In all cases, we report the strongest performance that we could obtain. All experiments are performed with the same random seed.

In Table 5.1, we present a comparison of the performance achieved by different network architectures in the simplified task scenario that does not involve pedestrians. It becomes evident that the Graph Neural Network (GNN) consistently outperforms

Comparison with other Network Architectures						
	Training Scenarios			Testing Scenario		
Agent	Coll.	Compl.	T.out	Coll.	Compl.	T.out
MLP	15	82	3	23	74	3
CNN	17	78	5	25	67	8
GNN (Ours)	8	92	0	13	85	2

Table 5.1: Comparison with other network architectures (baselines), measuring the percentage of completed episodes, collisions, and timeouts.

the other two architectures, namely the Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN), across various performance metrics. First and foremost, the GNN stands out in terms of the number of completed episodes, showcasing its superior ability to navigate and accomplish the given task successfully. This accomplishment implies that the GNN-based agents are better at learning and adapting to the environment, effectively completing more episodes within the same time frame compared to the MLP and CNN-based agents. This improved completion rate can be attributed to the GNN's inherent capability to capture complex relational information and dependencies among agents or entities within the environment.

Additionally, the GNN exhibits a notable advantage in reducing the number of collision episodes, which is a crucial metric in many multi-agent scenarios. Fewer collisions imply a safer and more efficient performance, highlighting the GNN's superior ability to model and anticipate interactions among agents or objects within the environment. This advantage can be explained by the GNN's capacity to effectively model spatial dependencies in a dynamic environment, which allows it to make better-informed decisions and avoid collisions more successfully.

Ablation Studies

Ablation studies have been conducted to investigate the effects of the node features (Eq. 5.1) and the addition of the route information and ego-agent speed information to the MDP state. In this Section we present out ablation studies to demonstrate the advantages of considering a) vehicle route information, b) vehicle speed. The experiments consist of a simplified intersection where no pedestrians are present.

The outcomes of our ablation studies, which scrutinize the influence of different input features on the policy network, are documented in Table 5.2. We systematically explore various combinations of input features, encompassing graph-based representations, route information, and ego-agent speed data. Importantly, we exclude combinations where the graph input is omitted, as they lack critical traffic-related information and are thus not relevant to our analysis.

The first row of Table 5.2 corresponds to a policy network leveraging only the spatial features of the graph as inputs. When a new input feature is added, the neural network architecture is modified so as to include an additional module that

Ablation Study on Network Inputs						
	Training Scenarios			Testing Scenario		
Agent	Coll.	Compl.	T.out	Coll.	Compl.	T.out
GNN	20	80	0	25	75	0
GNN Route	13	87	0	16	84	0
GNN Speed	16	79	5	24	76	0
GNN Route + Speed	**8**	**92**	**0**	**13**	**85**	**2**

Table 5.2: Ablation studies on network inputs.

Node Features						
	Training Scenarios			Testing Scenario		
Node features subset	Coll.	Compl.	T.out	Coll.	Compl.	T.out
x_t^n, y_t^n	19	74	7	21	70	9
$x_t^n, y_t^n, v_{x,t}^n, v_{y,t}^n$	11	84	5	13	79	8
$x_t^n, y_t^n, \theta_t^n, \lvert v\rvert_t^n$	14	81	5	16	77	7
$c_n, x_t^n, y_t^n, v_{x,t}^n, v_{y,t}^n$	12	84	4	12	82	6
$x_t^n, y_t^n, v_{x,t}^n, v_{y,t}^n, d^n$	**8**	**92**	**0**	**13**	**85**	**2**

Table 5.3: Ablation study on the node features of the graph.

processes the newly added information. The output from this module is seamlessly integrated into the feature fusion module, as depicted in Fig. 5.4.

As evident from the table, the introduction of both route information and ego-agent speed data leads to improvements in network performance. Of particular significance is the impact of route information, which results in substantial enhancements in both collision avoidance and task completion rates. Route information plays a pivotal role by providing insights into potential interactions with surrounding agents, thereby aiding the ego-vehicle in making more informed decisions.

Conversely, the inclusion of ego-agent speed information demonstrates a trade-off effect. While it effectively reduces collision rates, it may lead to an increase in timeout scenarios. This suggests that the knowledge of ego-vehicle speed is crucial in determining when to halt to avert collisions, but it can also result in more conservative decision-making, potentially leading to timeouts.

Ultimately, the most remarkable performance is achieved when all three input types described in Section 5.1.2 are considered simultaneously : graph spatial features, route information, and ego-agent speed. Remarkably, the performance on test scenarios only marginally declines, with a mere 5% reduction in both collision rates and task completion rates. This indicates that the network possesses the capacity to grasp the intricacies of intersection dynamics and generalize its learnings effectively to unseen tasks. Ablation studies on the node features used as input for the Graph Processing module have also been performed and the results are highlighted in Table 5.3.

We explore different feature combinations in this ablation study from the feature set in Eq. 5.1. We always include the agent position amongst the set of features as it is a crucial feature for this task. Adding velocity information in the form of linear velocity components $[v^n_{x,t}, v^n_{y,t}]$ or orientation and absolute velocity$[\theta^n_t, |v|^n_t]$ leads to improvements in the network performance compared to the position-only case. The performance of the set $\{x^n_t, y^n_t, v^n_{x,t}, v^n_{y,t}\}$ is slightly better than $\{x^n_t, y^n_t, \theta^n_t, |v|^n_t\}$, therefore additional features are added to former set. The addition of the minimum distance from the closest vehicle seems to have the major improvement in network performance. We argue that this is because the minimum distance plays an important role in deciding when the vehicle should stop to avoid a collision.

5.2.3 Experiment 2: SVO Effect on Mutual Interaction

Description

In this Section we explore the benefits of adding Social Value Orientation in the ego-agent's reward function. The scenario of this experiment is the following: The ego-vehicle will have to merge and overcome the intersection described in tasks a b and c of Figure 5.2. We also consider the presence of a pedestrian in the scene. We train both pedestrian and vehicle with game theory. The Level-0 pedestrian policy consists of a rule-based system: if the Time To Collision (TTC) of vehicles along the main road is more than 4 s from both sides, then the pedestrian will initiate crossing. We use the procedure described in Section 5.1.5 and Algorithm 2, to train Pedestrians of Level 1 and 2, and ego-vehicle policies Level 1 and 2, with SVO values in the set $\{0, 10, 20, 30, 40, 50, 60, 70, 80\}$. We do not consider purely altruistic agents (i.e. SVO 90) as they are not relevant to the Autonomous Driving task, as all agents at an intersection have at least the goal of reaching their destination.

The ego-vehicle performances are evaluated on the training and testing scenarios. The pedestrian policy at evaluation time is chosen randomly at the beginning of each episode.

Setting of Algorithm Parameters and Reward Function

Table 5.4 shows the parameter settings of the DDQN algorithm. To speed up the training process, the ϵ value for the ϵ-greedy policy is set to 0.9 for learning Level-1 policies and then set to 0.5 for Level-2 policies as in [171]. The same holds true for the number of training steps N_{training} and the replay memory buffer's size. The parameters of the ϵ-greedy action selection decreases from its initial value ϵ_0 linearly according to the training steps up to a minimum of ϵ_{min}. Let t indicate the current training step and ϵ_{decay} the training step at which the current learning rate ϵ reaches its minimum value, then ϵ can be computed according to the following expression:

$$\epsilon = \max\left(\epsilon_0\left(1.0 - \frac{t}{\epsilon_{\text{decay}}}\right), \epsilon_{\text{min}}\right) \tag{5.6}$$

Hyperparameters Settings	
Parameter	**Value**
Learning rate	1e-5
Starting (ending) value of ϵ greedy policy	0.9/0.5 (0.05)
Training Steps (N_{training})	400000/200000
Replay Memory Size	200000/100000
Number of actions	3
Mini-batch size	512
Discount Factor	0.99
ϵ_{decay}	$N_{\text{training}}/2$
Number of steps to update the target network	1000

Table 5.4: Parameters for multi-agent training

Reward Function Parameters	
SELFISH	
Goal reached r_g	0.5
Collision r_c	-2.0
Timeout r_t	-1.0
Velocity α_v	0.05
PRO-SOCIAL	
Velocity α_v^p	1.0

Table 5.5: Reward Function hyperparamters.

The reward function parameters are shown in Table 5.5. These parameters have already been discussed in Section 5.1.4. Here we will discuss them in greater detail. We report the reward function expression for clarity:

$$r(s_t, a_t) = \cos(\varphi) r_{self}(s_t, a_t) + \sin(\varphi) r_{others}(s_t, a_t) \tag{5.7}$$

with $r_{self}(s_t, a_t)$:

$$r_{self}(s_t, a_t) = r_c + r_t + r_g + \alpha_v r_v \tag{5.8}$$

If the ego-agent successfully reaches the destination, it receives a reward of 0.5 points. In the event of a collision, the ego-agent incurs a negative penalty of -2.0 points. Failure to move before the timeout window expires (set at 40 s) results in a negative penalty of -1.0 points. The parameter α_v governs the weighting of the speed reward term r_v, which is calculated as follows, based on the ego-agent's current velocity v and the agent's maximum velocity $v_{\max}$:

$$r_v = \begin{cases} 1.25 \cdot v/v_{\max} & \text{if} \quad v < 0.8 \cdot v_{\max} \\ 1.0 \quad \text{if} \quad v \geq 0.8 \cdot v_{\max} \text{ and } v < v_{\max} \\ 6.0 - 5.0 \cdot v/v_{\max} & \text{if} \quad v \geq v_{\max} \end{cases} \tag{5.9}$$

SVO	Coll.	Compl.	T.out	Av. Ep. Len.
0°	15/21	85/72	0/2	9.1 s
10°	20/22	79/75	1/3	5.7 s
20°	9/14	88/77	3/9	12.2 s
30°	11/16	88/79	1/5	13.7 s
40°	9/14	**91/83**	0/3	12.6 s
50°	9/12	**91/86**	0/2	15.3 s
60°	10/15	84/76	6/9	18.1 s
70°	13/17	79/72	8/11	20.0 s
80°	12/18	62/54	26/28	30.03 s

Table 5.6: Performance metrics for the ego-vehicle L2 policies against pedestrian policies. The first number in each entry of the table refers to the performance on the training scenarios, whereas the second number on the testing scenario. The last column reports the average episode length.

The value of v_{max} is set to match the road speed limit for the ego-vehicle and to 2.0 m/s for the pedestrian, approximately 1.5 times the average pedestrian crossing speed [228]. The rationale behind Equation 5.9 is as follows: when the current speed v is significantly below the maximum speed, the reward encourages the ego-agent to accelerate. If the ego-agent's speed is close to $v_{\max}$ but still below it, the ego-agent receives the maximum reward of 1.0 point (weighted by α_v). Finally, if the ego-agent's speed exceeds $v_{\max}$, the speed reward term begins to decrease until it becomes negative for speeds exceeding 1.2 times $v_{\max}$.

The altruistic reward term was introduced in Equation 5.4. The α_v^{other} is indicated in Table 5.5. Since surrounding agents reward function is weighted by the distance from the ego-agent, α_v^{other} is set to be equal to 1.0. In this way, the surrounding agent's reward function assumes values that are comparable with other reward terms (i.e. if it was set to 0.05 as α_v, the speed reward term for other agents would be negligible).

Simulation Results

We trained a total of 9 ego-agent policies for both the vehicle and the pedestrian for Level-1 and Level-2 agents. Here we analyse the effects of modelling a social term in the ego-vehicle's reward function.

The results are summarised in Table 5.6. The two values indicated in the table entries refer to the performances in the training scenarios (a,b) and testing scenario (c) respectively. Interestingly, the best performance according to the metrics is obtained for SVO value of 50°. Excellent performance is also obtained with SVO value of 40°. The number of completed episodes is higher for these two SVO values compared to completely selfish agents (SVO 0°) or altruistic agents (80°). Lower SVO values lead to an increase in collision rates, with the highest number of collisions occurring at an

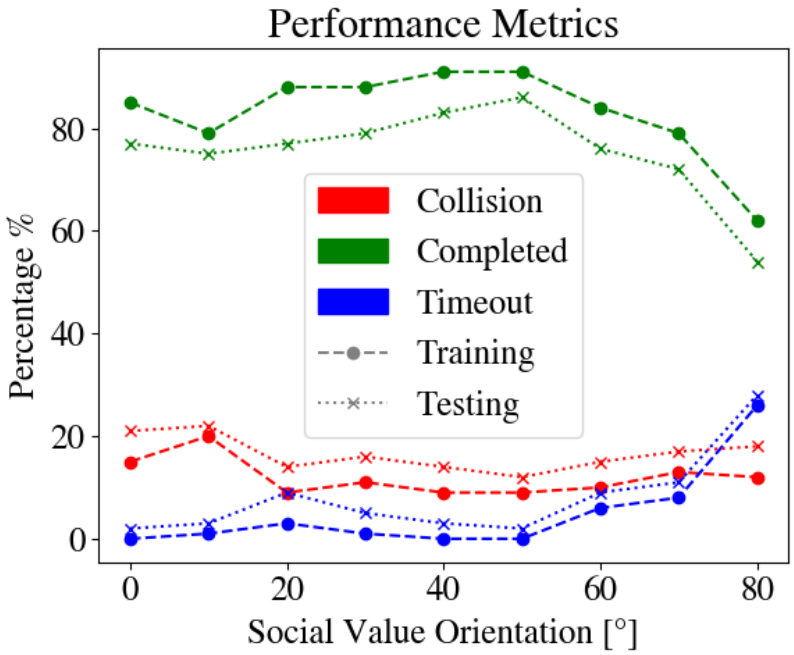
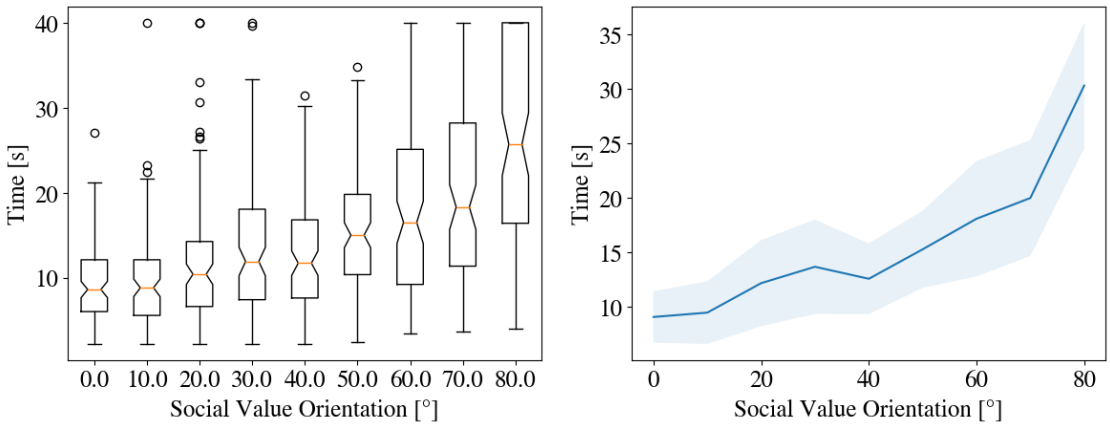


Figure 5.5: Performance metrics (collision, completed, and timeout) based on SVO value.

SVO value of 10°. This indicates that the addition of SVO can potentially lead to better ego-vehicle performance. On the other hand, excessive SVO values, e.g. 80° lead to a drastic increase in timeout rates. This is because the best policy for the ego-agent is to stand still and wait other vehicles to pass by. However, this is not an acceptable behaviour as the ego-vehicle might actually impede traffic flow, especially in scenario c (see Figure 5.2). The last column of the table indicates the average episode length.

The performance metrics trend is highlighted in Figure 5.5. As discussed in the preceding paragraph, it becomes evident that the timeout rate exhibits an upward trend with an increase in Social Value Orientation (SVO). This is primarily because ego-vehicles with higher SVO values tend to exhibit less haste in completing their tasks, resulting in a higher likelihood of timing out. Interestingly, the decline in the number of completed episodes for the ego-vehicle as SVO increases is closely associated with the rise in timeout occurrences rather than collision rates. Collision rates remain relatively stable throughout, with a noticeable decrease occurring only when SVO surpasses 20°. These results are in line with the results obtained in Chapter 4 for the single-agent case.

It is noteworthy that despite the heightened complexity of the task compared to Experiment 1, the ego-vehicle still manages to maintain a comparable rate of completed episodes. This indicates that the ego-vehicle's adaptability and performance are robust, even in more demanding scenarios.



(a) (b)

Figure 5.6: a) Box and Whisker plot for average episode length. b) Average episode length with standard deviation error.

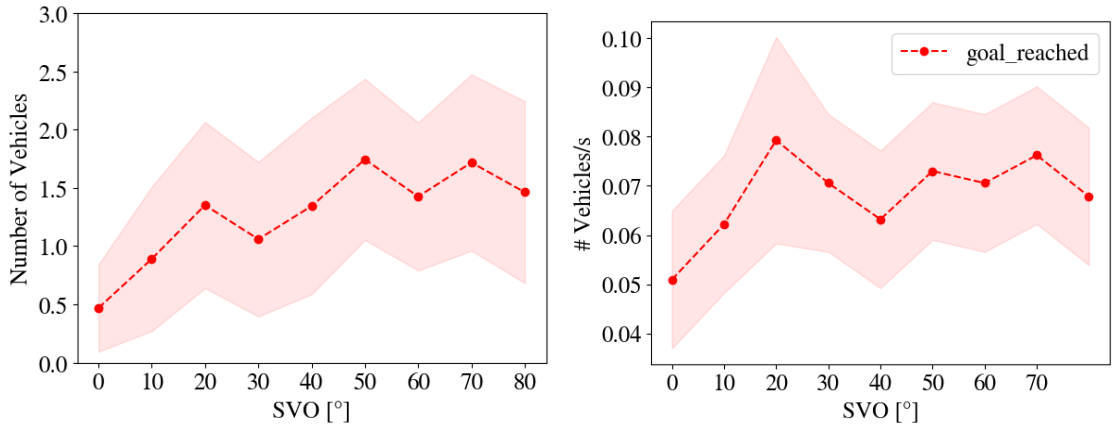
The average episode length is also correlated with the SVO value, as shown in Figure 5.6a and 5.6b. This correlation shows how an individual's SVO influences the behavior of the ego-vehicle in the multi-agent environment.

Specifically, the average episode length exhibits an upward trend with increasing SVO values. This pattern can be attributed to the fact that, as an ego-vehicle's SVO becomes more oriented towards cooperation and yielding, it tends to engage in behaviors that prioritize social harmony over swift task completion. Consequently, episodes take longer to conclude, reflecting the ego-vehicle's propensity to yield the right of way to other agents in the environment.

The spread in episodes duration shown in Fig. 5.6a also increases with SVO values. This dispersion is primarily a consequence of the inherent variability in the number of vehicles present in the intersection during different simulation runs. In scenarios where the intersection is devoid of other vehicles, the ego-vehicle can swiftly complete its episode, regardless of its SVO value, resulting in shorter episode durations. Conversely, when the intersection is crowded with other agents, the ego-vehicle's cooperative tendencies are more likely to manifest, leading to longer episode durations. This phenomenon contributes to the observed variability in episode lengths and is a testament to the dynamic nature of the multi-agent environment. Since the vehicle is less likely to yield and will try to complete its task swiftly for low SVO values, this phenomenon is less likely to occur, resulting in a lower variance for small SVO values.

Importantly, these findings are in alignment with the results obtained in Chapter 4, suggesting a degree of consistency and robustness in the impact of SVO on the ego-vehicle's decision-making across various experimental setups and scenarios.

In summary, the correlation between SVO values and average episode length provides valuable insights into how an ego-vehicle's social disposition affects its behavior in multi-agent environments, highlighting the intricate interplay between coopera-



(a) Number of Vehicles crossing the intersection before the ego-vehicle in each episode.

(b) Number of vehicles that cross the intersection per second.

Figure 5.7: Effect of Social Value Orientation on traffic flow at the analysed intersection.

tion, episode duration, and the presence of other agents. These findings underscore the nuanced challenges of developing interaction-aware decision-making strategies for autonomous vehicles.

Effect on Traffic Flow

The impact of the introduction of Social Value Orientation in the reward function is shown in Figure 5.7. In Figure5.7a, we can observe how the average number of vehicles crossing the intersection is influenced by the ego-vehicle's behavior. The altruistic policy, which prioritizes allowing other vehicles to cross before itself, naturally leads to a situation where an increasing number of vehicles can smoothly navigate the intersection. This behavior aligns with the principles of social cooperation and contributes to a more harmonious and efficient traffic environment.

Figure 5.7b specifically focuses on the average number of vehicles crossing the intersection per second, excluding the ego-vehicle from the calculation. Here, the introduction of the social reward term into the reward function is shown to have a positive impact, with an increased rate of roughly 30% in our simulations. The increased intersection crossing rate demonstrates that an altruistic policy not only enhances safety by reducing collision rates but also has the potential to bring about significant advantages in terms of traffic flow dynamics. This insight suggests that promoting altruism among autonomous vehicles can lead to smoother and more efficient traffic patterns, ultimately benefiting both the individual vehicle and the overall transportation system.

5.3 Conclusions

In this Chapter, we have developed a framework that allows to extend Social Value Orientation to multi-agent settings. We have analysed the advantages of using GNN as policy network in DRL and demonstrated its superior performances compared to other common Neural Network architecture on this Autonomous Driving task.

The GNN's superior performance can be attributed to its unique architecture, which leverages graph-based representations to capture complex relationships and dependencies among entities in the environment. Unlike the MLP and CNN, which may struggle to model such intricate interactions, the GNN excels in handling scenarios involving multiple agents and their intricate interplays. It achieves this by considering the environment as a graph, where nodes represent agents or entities, and edges represent the relationships between them. This graph-based approach enables the GNN to efficiently propagate information and make informed decisions by considering the context of neighboring entities. In summary, the GNN outperforms the MLP and CNN networks in this simplified multi-agent task due to its inherent ability to capture complex relational information, model interactions among agents, and make better-informed decisions. This demonstrates the strength of graph-based neural networks in scenarios where understanding and leveraging relationships among entities in the environment are critical for achieving superior performance.

Ablation studies have been conducted to find optimal features to solve the task. Our ablation studies on input features underscore the critical role played by route information and ego-agent speed data in enhancing the performance of our model within a simplified intersection environment. These findings shed light on the importance of considering contextual information when modeling multi-agent systems, ultimately contributing to more informed and effective decision-making in complex traffic scenarios.

We have introduced a novel multi-agent social reward function, which enables us to consider the impact of Social Value Orientation (SVO) on the behavior of multiple agents within the traffic environment. The results obtained from our study demonstrate that the incorporation of SVO into the reward function of the ego-vehicle yields promising advantages for enhancing the overall traffic conditions and interactions.

In this work, we mainly focused on a quantitative analysis of the impact of SVO on traffic. Future work is essential, for example, to validate the human-likeness of the AV policy . Chapter 6 will primarily concentrate on the development of Virtual Reality validation systems designed for testing Autonomous Vehicle (AV) algorithms. Our next step is to build up a Virtual Reality data capturing environment and perform a subjective human factor analysis with a human-in-the-loop to evaluate the policy.

Another critical concern pertains to the collision rates achieved by the DRL agent. Given that it did not consistently reach zero across all episodes and Social Value Orientation (SVO) values, it is imperative to integrate an additional safety layer. This additional layer will ensure the safety of other road users. While the DRL

policy can still serve as a valuable guideline for guiding AV decision-making, the incorporation of a safety layer is imperative.

Lastly, to ensure the robustness of the policy in various conditions, it is necessary to include a broader range of road layouts and scenarios in the study. This expansion will help guarantee that the policy functions effectively under diverse circumstances.

Chapter 6

Virtual Reality for AVs

6.1 Overview

Pedestrian safety has always been a major concern on the roads. With the rise of autonomous vehicles (AVs), it becomes even more crucial to understand pedestrian behavior to ensure their safety. As stated in [229], pedestrians are at a high risk of accidents, accounting for over 22% of all road traffic fatalities in the European Union in 2013. Moreover, the behavior of pedestrians on the road is highly variable, depending on several factors such as their age, gender, culture, and even mood. For instance, children may be more prone to distraction while elderly individuals may walk more slowly and need more time to cross the road. Therefore, investigating pedestrian behavior and identifying patterns can help AVs adapt to different scenarios and minimize the risk of accidents. By analyzing pedestrian behavior, we can develop better technology that can prevent accidents and protect the most vulnerable road users. According to research in [35], the kinematics and signaling information of autonomous vehicles (AVs) play a crucial role in influencing pedestrian behavior, particularly since there is no driver involved. Therefore, it is important to identify the specific motion cues or signals that have the most significant impact on pedestrian behavior, as this holds significant research value.

Previous research has generally agreed that the distance or time to collision (TTC) between vehicles and pedestrians is the primary kinematic cue that influences pedestrian behavior [104] However, a recent study has shown that pedestrians use multiple sources of information from vehicle kinematics instead of relying solely on one cue. The impact of speed, distance, and TTC on pedestrian behavior is mutually coupled [58]. Moreover, in pedestrian-vehicle interactions, evidence suggests that the driving maneuver of the driver, such as deceleration, plays a critical role in affecting pedestrian behavior. Vehicle movements are also linked to pedestrian trust in vehicles, emotions, and influence [15].

Chapters 4 and 5 have introduced decision-making algorithms for AVs but have mainly focused on DRL studies in simulation scenarios. In order to bridge the gap

between simulation and reality multiple approaches are possible. AV technology is extremely costly and performing real world experiments on the road can be difficult, especially when VRUs are involved. Acquiring data to study pedestrian movement can be both difficult and costly. Although near-collision events are crucial to developing accurate pedestrian models, analyzing them in the real world can be dangerous. Fortunately, recent advancements in Virtual Reality (VR) technology have enabled the creation of virtual environments that provide a safe and cost-effective means of collecting data for Autonomous Driving (AD) studies [230].

In this work, we intend to develop a VR road simulator to study pedestrian behavior and decision-making. To study the reciprocal interactions between driver and pedestrian, the pedestrian will wear a VR headset that immerses them in a virtual environment. A human driver will sit in front of a screen and use a steering wheel and pedals to control the car. While this study focuses on the interactions between a single human driver and pedestrian, the system is designed so that it can easily be extended to more cluttered environments with multiple vehicles and pedestrians. The system is then validated with a data collection experiment and perform a deep learning based analysis of pedestrian motion. Although this study will mainly focus on data driven pedestrian models, the VR simulator developed here can serve as a testing ground for future AV algorithm or behavioural models testing.

6.2 Design Choices

(a) Logitech G920 Steering Wheel and Pedals.

(b) Perception Neuron Motion Capture System.

(c) HTC Vive Pro 2.

Virtual Reality has several advantages compared to real world tests in AD. Virtual Reality allows for a safe and cost effective study and data collection of interactions. A systematic review from [231] shows that the number of Augmented Reality and Virtual Reality papers with applications in AD has been increasing in the most recent years with an increasing interest in Vulnearable Road Users (VRUs), such as pedestrians or cyclists. VR has several advantages over field data collection for AD research. First of all, safety is ensured thanks to the fact that traffic participants interact within a virtual environments. Collision and near collision scenarios no longer constitute a problem for VRUs. The environment is entirely controllable, which can be useful for studying specific scenarios or testing different hypotheses. Besides, VR is a more

cost-effective way to collect data because it does not require the use of real vehicles and expensive sensors.

The discrepancy between Reality and Virtual Reality has been studied in [232]. In their work they showed that no statistical significant differences are present between the virtual and the real environments in pedestrians' intention to cross. The same also holds true for the perceived risk and safety of crossing and perceived distance between pedestrians and vehicles, and perceived vehicle speed. There were, however, statistically significant differences between the estimation of the speed of the approaching vehicle in the virtual and real environments. Pedestrians had higher estimations of speed in both the virtual environments than the real environments. The difference is perceived speed is also highlighted in [233].

An alternative to Virtual Reality Displays is HoloLens, which is a mixed reality headset that allows users to see digital content overlaid on the real world. It was first released in 2016 and is one of the most advanced HMDs on the market. HoloLens has been employed to by [234] to develop a pedestrian simulator AR-PED that allows the users total freedom of movement without any physical boundary restrictions. However, Hololens is a more expensive solution compared to VR displays, such as HTC Vive. For this reason, and because it also provides a more 3D immersive experience, it was decided to use HTC Vive Pro 2 headset over the Hololens.

This work only focuses on a pedestrian user interacting with an AI driven vehicle. In this work, we develop a pedestrian simulator using Unreal Engine 4.26 based on HTC Vive Head Mounted Display. The simulator allows two simultaneous users: a human driven vehicle and a pedestrian. The simulator can be used to carry out research on pedestrian-driver interactions, as well as for testing Autonomous Driving algorithms with humans in the loop in a safe manner.

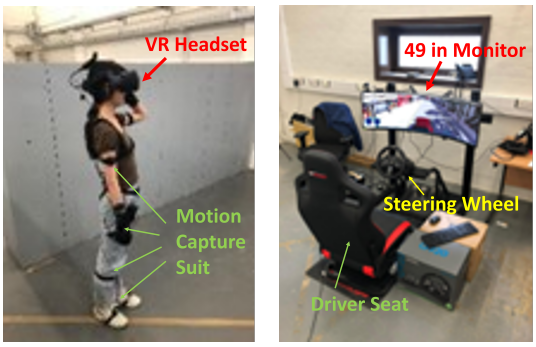
We then used state-of-the-art deep learning techniques to estimate the joint trajectory of both the pedestrian and the car. Deep learning is becoming more and more common in predicting pedestrian trajectories because of its impressive ability to represent data. In particular, the Social-LSTM [127] uses Recurrent Neural Networks (RNNs) to model the trajectory of each pedestrian in combination with a social-pooling operation to consider surrouding agents. Another approach in modeling human-human interaction for pedestrian trajectory prediction is to use graphs, as they can better capture the structure of the scene. Social-BiGAT [235] uses a combination of LSTM to model the trajectory of each pedestrian and Graph Attention Network (GAT) to model their interactions. In this work, it was decided to use STGCNN network, in order to be able to include multiple agents in the future, we used Social-STGCNN [236], which represents trajectories as a spatio-temporal graph.

The contributions of this Chapter are the following:

- the development of a traffic simulator, with a wireless HMD device (HTC Vive) that allows the users freedom of movement in combination with a motion capture system;
- a framework where pedestrians and user controlled vehicles can coexist with

each other, as well as with Autonomous Vehicles. This framework can be used in the future to aid Autonomous Driving research for validation and testing;

- the analysis of the pedestrian and driver motion with Deep Learning Techniques.



(a) Pedestrian. (b) Driver system.

Figure 6.2: Overview of hardware components.

6.3 Methodology

This study introduces a new simulation framework that allows users to control pedestrians and vehicles in a virtual environment. The framework is powered by the latest advances in computer hardware, software, and networking. The following sections will detail the system design and key components of the system. we also conducted a data collection experiment where we invited participants to evaluate the system and collected their trajectory data. VR allows developers to create realistic simulations of pedestrians in a variety of environments. This allows them to test autonomous vehicles in a safe and controlled environment, and to collect data on how pedestrians interact with autonomous vehicles. This data can then be used to improve the safety and performance of autonomous vehicles. The trajectory data will be analysed with deep learning techniques to develop a trajectory prediction model, which can aid the development of pedestrian simulators as well.

6.3.1 Design of the Virtual Environment

Developing a virtual reality environment requires to combine together a lot of different pieces of hardware and software. In this subsection, we will go through the system components and design. The current system allows a pedestrian and a user controlled vehicle access to the virtual environment at the same time.

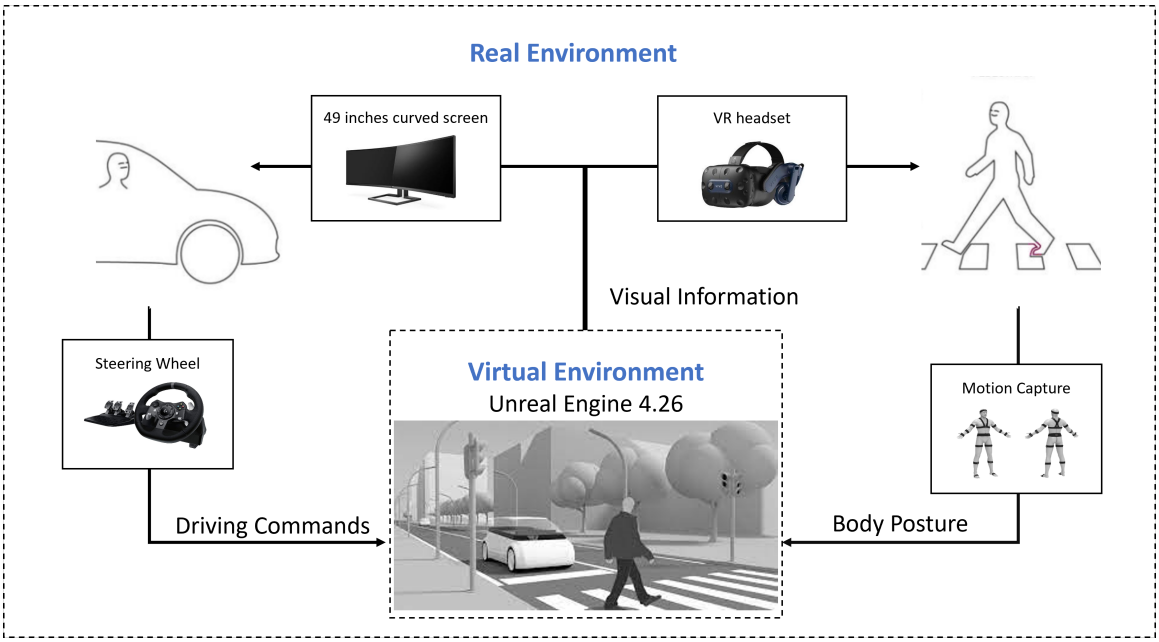


Figure 6.3: Overview of the VR Environment.

Hardware Components

The pedestrian hardware components consist of a Perception Neuron Motion Capture Suit (Fig. 6.1b) and an HTC Vive Pro 2 Virtual Reality headset (see Fig. 6.1c). The Perception Neuron Motion Capture Suit is a wearable system that tracks the position and orientation of the body's major joints. It obtains a pedestrian posture representation which is transferred to the virtual environment to create a realistic digital representation of the virtual pedestrian (pedestrian avatar). The HTC Vive Pro 2 VR headset is a high-end VR headset that provides a sharp, high-resolution image. It also has a wide field of view, which helps to create a more immersive experience. Figure 6.2a shows a participant wearing the mocap suit and the VR headset. The VR headset is mounting a wifi-adapter that allows the user to move freely, without being tethered to the desktop computer.

A 49 in screen is used to display the virtual reality environment to the driver. The driver is controlling the virtual vehicle with a Logitech G920 Steering Wheel and Pedals (see Fig. 6.1a). The Logitech G920 Steering Wheel and Pedals are a high-quality driving simulator set that provides realistic feedback to the driver. This helps the driver to feel like they are actually driving a real vehicle. A driver seat is also used to make the overall experience more similar to a real vehicle, resembling a driver video game. The driver seat can be adjusted to give a comfortable experience to different drivers. The current lab allows us to operate in an area of approximately 7 m $\times$ 7 m and the SteamVR Base Station allows an area of up to 10 m $\times$ 10 m. This effectively limits the area where the pedestrian can move, thereby impacting the

(a) Top view of the map.

(b) Single lane view.

Figure 6.4: Screenshots of the virtual environment.

maximum size of the virtual environment in which the virtual pedestrian can operate. For this reason, we focused on a single road environment in this first study and we will consider multi-lane scenarios when possible.

The VR environment runs on a desktop computer with an NVIDIA GeForce RTX 3060 Ti GPU and 11th Gen Intel(R) Core(TM) i7-11700 @ 2.50GHz GPU. To keep the costs low the driver and the pedestrian are connected to the same machine but it is also possible to have multiple machines connected to each other in a client-server architecture.

Figure 6.5: Example of data collection experiment, showing the VR user and the driver.

Software Components

The VR environment was designed with Unreal Engine (UE) 4.26 software, a powerful game engine that is used to create realistic and immersive virtual worlds. Unreal Engine 4.26 provides a wide range of features that are specifically designed for VR development and easily allows to integrate and develop VR games. The motion capture raw information is processed by Axis Neuron, the software provided by Perception Neuron. Axis Neuron allows to stream body posture to Unreal Engine 4.26 via TCP/IP connection and also provides an UE 4.26 plugin that can animate pedestrian avatars. The VR headset is also connected to UE 4.26 via SteamVR, a virtual reality platform. One of the main technical issues faced when combining the

VR headset with the motion capture system is given by the fact that they use two different reference systems. Both the VR headset and the Perception Neuron con stream pedestrian position information to unreal engine. We decided to rely to the VR headset head position and orientation and to use the motion capture system to animate the body posture of the avatar, since the VR has lower measurement noise (STD 1.0 cm for the headset and STD 3.0 cm for the motion capture system) and relies on a fixed coordinate system whereas the origin of the motion capture coordinate system depends on the calibration procedure. A practical problem that was encountered during the data collection is that the motion capture unit that is placed on the head is often giving wrong head orientation information due to the interference with the VR headset placed on top of it. Therefore, it was decided to rely on the VR system for head orientation as well.

The driving commands can be sent to UE 4.26 directly. However, this does not allow to change vehicle dynamic parameters and does not give access to dynamical information such as vehicle acceleration, angular acceleration or friction. We decided to simulate the vehicle dynamics with a Python script, which captures the driver commands and updates the vehicle position in UE 4.26. UE 4.26 provides information such as ground surface structure and vehicle model parameters that allow the Python Script to update the vehicle dynamics.

Map Design

As already mentioned, we used Unreal Engine 4.26 to develop a virtual urban environment. We are interested in studying interactions with a vehicle and a pedestrian when the pedestrian is attempting to cross the road without any road signs (no zebra crossing or traffic lights). We designed a single lane environment where with a loop-shaped road structure (see Fig. 6.4a). The shape of the map will be very useful during the data collection experiments, as it allows the driver to constantly driving without having to restart the simulation after each interaction episode is over. We built a 3.65m wide single-lane road (see Fig. 6.4b) and added some buildings and trees to make it resemble a realistic road. No obstacles were added near the pavement as the effect of occlusion on driver and pedestrian decisions are beyond the scope of this study. After setting up the whole simulation system, drivers sitting in front of the driving simulator can see the pedestrian in the virtual traffic environment on the screen while the pedestrian can see the car operated by the driver in the VR headset.

6.3.2 Trajectory Prediction

Trajectory prediction is important for autonomous driving because it allows the vehicle to anticipate the movements of other traffic participants, such as vehicles, pedestrians, and cyclists. This information is essential for the vehicle to make safe and timely decisions, such as when to brake, change lanes, or accelerate. We use the network proposed by [236], which consists of Spatio-Temporal Graph Convolutional Neural

Network (ST-GCNN) layers and Time-Extrapolator Convolutional Neural Network (TXP-CNN) layers, with the Parametric ReLU (PReLU) activation functions.

The problem can be formulated as follows: we have a group of N agents (vehicles and pedestrians) whose trajectory is observed over a time period T_o, and we want to predict their future trajectories over a time horizon T_p. For each agent n, we represent their trajectory as a set of 2D coordinates (x_n^t, y_n^t) for each time step t. We assume that the distribution of these coordinates follows a bi-variate Gaussian distribution, denoted as $p_n^t \sim \mathcal{N}(\mu_n^t, \sigma_n^t, \rho_n^t)$. Our goal is to estimate the parameters of this distribution $(\mu, \sigma, and \rho)$ and minimize the negative log-likelihood to improve the accuracy of our predictions. We denote the predicted trajectory as $\hat{p}_n^t$, which follows the estimated bi-variate Gaussian distribution $\mathcal{N}(\hat{\mu}_n^t, \hat{\sigma}_n^t, \hat{\rho}_n^t)$. The model is trained to minimize:

$$L^n(\mathbf{W}) = -\sum_{t=1}^{T_p} \log\left(\mathcal{P}\left(\mathbf{p}_t^n | \hat{\mu}_t^n, \hat{\sigma}_t^n, \hat{\rho}_t^n\right)\right), \tag{6.1}$$

where $\mathbf{W}$ are all the trainable parameters of the model, $\hat{\mu}_t^n$, $\hat{\sigma}_t^n$, $\hat{\rho}_t^n$ are the mean, variance and correlation of the distribution.

We have adapted the network architecture proposed in [236] to better suit our problem. In particular, the network has been designed to predict pedestrian motion in crowded scenarios. We chose this network architecture because it has shown excellent performance in trajectory prediction tasks and can be easily extended to include more pedestrians and vehicles in the future. Choosing a neural network architecture for its ability to accommodate multiple agents is a strategic decision that prepares for complex scenarios involving multiple entities. This architecture offers flexibility for future scenarios and aligns with the goal of studying multi-agent AV-pedestrian interactions.

Since in our scenario only one vehicle and one pedestrian are present, thereby consisting of fewer agents compared to [236], overfitting of the available data is a significant problem. To overcome this issue, we have modified the loss in 6.1 by adding a regularisation term. We have introduced data augmentation by applying transformations on the input data, including rotations and reflection. This allows to effectively increase the available data without loss of generality. Ablation studies on the network layers and hyperparameters are highlighted in Section 6.4.2.

6.4 Experiments

6.4.1 Data Collection

We collected interaction trajectories of the pedestrian and the car in a virtual environment. For the driver, we collected the steering, throttle, and brake commands, as well as the car's absolute position. Pedestrian data consisted of their absolute position within the virtual environment. The objective is to predict the future behavior

of the pedestrian based on the behavior of the car. Unlike other datasets present in literature, this dataset contains not only relative position data, but also commands used by the driver. For example, if the driver suddenly brakes, the pedestrian may be surprised and change their course of action. This information is not captured by datasets that only contain relative position data.

Figure 6.6: Overview of the VR Environment.

Fig. 6.5 shows the process of the data collection experiment in the real world and Fig. 6.6 shows the corresponding scenes in virtual reality. The pedestrian can spawn at any point along the straight road segments and on both sides of the road. This has two main advantages. Firstly, the driver is unaware of the exact pedestrian spawn location, which adds uncertainty to the scenario. Secondly, the data collected has a bigger variety, which can improve generalisation of machine learning methods, such as those used in Section 6.4.2. After setting up the whole simulation system, drivers sitting in front of the driving simulator can see the pedestrian in the virtual traffic environment on the screen while the pedestrian can see the car operated by the driver in the VR headset, as shown in Fig. 6.5.

During the recordings, the pedestrian was told to make a crossing decision every time they saw the car coming. Since the relative initial distance between the car and the pedestrian is random, the data has a wider variety of initial Time To Collision (TTC) values. TTC is a measure of how much time it will take for two objects to collide. It is a critical metric for autonomous vehicles, as it allows them to assess the risk of a collision and take evasive action if necessary.

A total of 16 driver-pedestrian pairs was invited to take part to the study. 775 interaction episodes were collected. Out of these, 480 were considered valid after post-processing. Among the 480 events, we observed that pedestrians decided to cross the road in 242 cases, which makes the dataset balanced between crossing and

	Car	Pedestrian	Average
No weights	3.10/5.81	0.36/0.55	1.73/3.18
L1	0.93/1.30	0.18/0.30	0.55/0.80
L2	**0.85/1.39**	**0.17/0.25**	**0.51/0.83**
Learnable	0.86/1.38	0.19/0.26	0.53/0.82
MLP	1.13/1.62	0.33/0.65	0.73/1.14

Table 6.1: Network performance based with different adjacency matrix. No weights refer to an adjacency matrix with ones on the diagonal, L1 and L2 norms are also analysed.

non-crossing action. The participants are people of different ages and genders. The drivers are all people with at least 3 years of driving experience, holding a valid UK driving license. Data for each pair was recorded for about an hour, with an average recording time of 30 minutes for each pair. The total is 8 hours of effective trajectory data for driver-pedestrian interactions. The data collected has then been pre-processed for neural network training.

6.4.2 Experimental Results

We use the network described in [236], consisting of ST-GCNN layers and TXP-CNN layers, with the PReLU activation functions. We chose this network as it has demonstrated excellent capabilities in the trajectory prediction tasks and can easily be extended in the future to include more pedestrians and vehicles. We divide the collected data into three groups: training, validation, and test datasets with roughly a 7:2:1 ratio (i.e. 11 driver-pedestrian pairs for training, 3 pairs for validation and 2 pairs for testing, chosen randomly).

We utilized a training batch size of 32 and employed Stochastic Gradient Descent (SGD) to train the model for 250 epochs, setting the initial learning rate to 0.01 with linear decay. The choice of batch size was influenced by the size of the dataset and compared against network performances obtained with batch sizes of 16 and 64. One of the major problems encountered during training on our dataset was overfitting. To mitigate the risk of overfitting in our neural network model, we have incorporated dropout and regularization loss into our training process. These techniques collectively serve as effective safeguards, enhancing the generalization capabilities of our model and ensuring its performance on unseen data. Ablation studies were also conducted on the number of STGCNN and TXP-CNN layers. As reported in Table 6.1, the optimal number of layers was found to be 2 STGCNN layers and 5 TXP-CNN layers. Higher number of layers (3 STGCNN) resulted in network performance degradation due to rapid overfitting. On the other hand a single STGCNN layer might not be enough to capture complex human interactions, therefore resulting in worse performances.

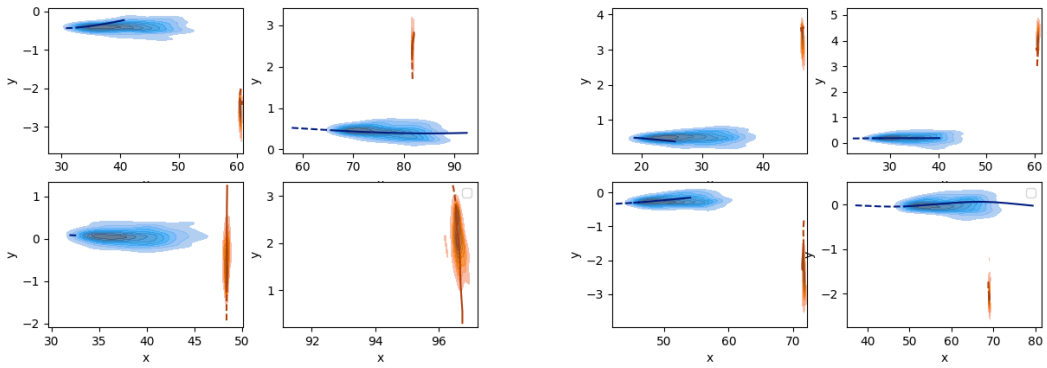


Figure 6.7: Some pedestrian (orange) and car (blue) trajectories with predictions. Previous trajectory (dashed line), future trajectory ground truth (solid line), the color density is the predicted trajectory distribution.

We conduct ablation studies on the adjacency matrix function. We consider no-weights, the reciprocal of L1 distance, the reciprocal of L2 distance, and a learnable adjacency matrix. Given the simplicity of our prediction task, consisting of trajectories of only two agents, we do not see any considerable improvements with the learnable adjacency matrix. No-weights refers to an unweighted adjancency matrix. We see the best results using the L2 distance, which captures the spatial relationships between the agents. In particular, this result confirms the intuition that the further away the agents are from each other, the less the mutual influence is. The time horizon for the prediction is set to 2.4 s in the future, as this is considered long term prediction for autonomous driving tasks. The performance of our network is shown in table 6.1. The metrics used to evaluate the trajectory prediction model are the Average Displacement Error (ADE) and Final Displacement Error (FDE). ADE is the average distance between the predicted trajectory and the ground truth trajectory over the entire prediction horizon. FDE is the distance between the predicted trajectory and the ground truth trajectory at the end of the prediction horizon. Table 6.1 shows the network performance on the task with the adjacency matrix related to the studies. We also compare the graph-neural network to a multi-layer perception network. So far, the deep neural network analysis demonstrates that the network is capable of predicting pedestrian and car future trajectories. We are looking forward to collecting more data and including skeleton data to release an open-source dataset for deep learning that is based on VR collected data.

Fig. 6.7 shows some sample trajectories for our prediction task. The pedestrian (orange) is crossing in the vertical direction, whereas the car (blue) is moving in the horizontal direction. The predicted future trajectory distribution is represented with a coloured density. Our method's trajectory predictions work well, showing that it's a good fit for our problem. The STGCNN layers ensure that the interactions between the driver and the pedestrian are learnt by the network, outperforming a simpler

MLP network. The sample trajectory predictions show that the method effectively predicts the probability of future positions for the car and the pedestrian.

6.5 Conclusions and Discussions

In this work, we introduced a Virtual Reality environment and a data collection framework for pedestrian trajectories, which ensures safety and limited costs for the experiment. The simulator can be used in many different Autonomous Vehicles research opportunities in the area of AV/driver interactions with pedestrians, testing of AV control algorithms, pedestrian behaviour prediction and safety. We then analysed the collected data with a motion prediction system based on deep learning which demonstrates that the system can used to predict trajectories for pedestrians that are not present in the dataset.

Future research directions will have to include multi-sensor data, as we have only focused on pedestrian-car 2D trajectories, neglecting camera or pedestrian pose for making predictions. Including these type of data might be useful for improving crossing probability estimation. The current virtual environment setup is only suitable for one driver and one pedestrian but it can be extended for the study of multi-drivers and multiple pedestrians. Gap-acceptance studies for multiple-lanes scenarios is also a possible future research directions, especially by employing AI-driven vehicles in the scene. This is an easy way to test how pedestrians would behave in a multiple-lane scenarios, which has not been researched extensively in literature, due to the costs of setting up such experiments. How to model drivers' and pedestrians' behaviour in such scenarios is still an open research question.

The virtual reality (VR) environment developed and described in this chapter will play a crucial role in future research, particularly in relation to the simulations and car reinforcement learning models introduced in chapters 4 and 5. Specifically, we aim to leverage this VR platform to provide a controlled and risk-free environment to validate the vehicle models. By conducting these tests in a virtual setting, we can rigorously assess the performance, safety, and decision-making capabilities of the reinforcement learning algorithms without posing any threat to pedestrians or other real-world entities. This approach not only enhances the robustness of the models by allowing for extensive testing under various scenarios and conditions but also ensures that the learning process remains entirely safe and ethical, free from the potential hazards associated with real-world experimentation.

An open research question that arises from this work is how to ensure that the data collected in VR resembles that of real world. One of the key challenges in VR is creating virtual environments that closely mimic real-world environments. Researchers have been exploring ways to improve the fidelity of VR simulations to reduce the mismatch between virtual and real-world data [237]. This includes efforts to create more realistic graphics, physics simulations, and interactions. Reducing the data distribution mismatch between VR and real world data would allow to drastically reduce

costs, ensure safety, and speed up testing of new AV technology.

Chapter 7

Conclusions

This thesis has addressed the utilisation of Social Psychology aspects in the Deep Reinforcement Learning framework to improve the quality of decisions made by an Autonomous Vehicles. Despite the significant utilisation of DRL technology for Autonomous Driving, most studies neglect social aspects of driving in the policies. The aim of this thesis was to explore the potential benefits of including social aspects into the design of AV policies. In particular, we have focused on the integration of Social Value Orientation in the DRL framework and analysed its effects on learning-based decision-making policies.

This thesis has explored the impact that SVO can make when learning Autonomous Driving policies in social contexts. SVO has been tested in a two agents framework, as well as in multi-agent settings. SVO has been employed with different neural network architectures, namely multilayer perceptron (MLPs), Convolutional Neural Networks (CNNs), and Graph Neural Networks (GNNs). SVO has proven to be a flexible and general framework that can improve the vehicle impact on traffic. The benefits of including SVO in traffic flow have been analysed in Chapters 4 and 5, where we have shown how the rate of vehicles passing through an intersection can improve. Our multi-agent analysis also showed that including SVO in multi-agent scenarios reduces the collision rates obtain by DRL policy. However, in order to deploy the DRL agent into the real world, lower collision rates must be obtained. We suggest including an additional safety layer to prevent car crashes that act as a safety layer for the DRL agent. Furthermore, the idea of utilising deep learning approaches to learn AD policies can lead to more general systems that require fewer handcrafted rule-based systems in the AV decision-making and controller design.

SVO has been shown to improve DRL agents performance in both single agent scenarios and also in multi-agent scenarios. By leveraging social factors in the reward function design of the MDP formulation, the ego-vehicle naturally exhibits human-like behaviour such as early stopping and yielding in favour of surrounding agents. However, it is acknowledged that further research is needed to precisely calibrate the degree of human-likeness inherent in these behaviors.

A Virtual Reality (VR) environment was developed for pedestrian trajectory data

collection, ensuring cost-effectiveness and safety. The VR simulator offers versatile applications in Autonomous Vehicle (AV) research, including AV-pedestrian interactions, control algorithm testing, and pedestrian behavior prediction. Analysis using deep learning-based motion prediction demonstrated the system's ability to forecast trajectories for pedestrians and vehicles. Future research should explore multi-sensor data incorporation and expand the virtual environment for multi-driver and multi-pedestrian studies. Gap-acceptance investigations in multi-lane scenarios, particularly with AI-driven vehicles, represent promising research avenues. Future plans will involve using the developed VR environment to test AV policies in a safety environment. In particular, social factors studies with subjective evaluation can be performed in a safety environment. In this way, the VR environment will act as a first cost effective testing ground for AV policies before they can be deployed and tested in the real world. The VR environment can also be used to study pedestrian and driver behaviour within a safe and controlled environment.

7.1 Thesis Contributions

The main contributions of this thesis are:

- A novel conceptual framework to integrate social psychology into the AV controller design. In particular, the introduction of SVO into the DRL framwork to influence the ego-vehicle strategies, achieving behaviours that range from egoistic to pro-social. We designed tests to evaluate the impact of Social Value Orientation in two-agent and multi-agent scenarios, highlighting the impact of SVO on traffic flow and agent performance.

- We extended the aforementioned SVO framework to multi-agent scenarios by introducing a novel Graph Neural Network (GNN) architecture. This GNN architecture combines route, traffic, and ego-vehicle information to make decisions, enabling the SVO framework to reason about the behavior of other agents in the environment and coordinate its actions accordingly.

- We proposed a new pedestrian model for computer simulation that combines gap-acceptance and social-force models, with the addition of a situational awareness risk assessment to trigger crossing. We showed that the DRL agent can still handle more complex human models, which is essential for simulating real pedestrians.

- We developed a Virtual Reality (VR) traffic simulator, paired with a wireless HMD device (HTC Vive), that allows users to move freely and interact with pedestrians, user-controlled vehicles, and autonomous vehicles. This VR framework can be used in the future to aid autonomous driving research for validation and testing, enabling researchers to assess the performance of autonomous vehicles in a variety of realistic traffic scenarios.

- We used the VR traffic simulator to collect interactive trajectory data between a human-driven vehicle and a pedestrian. We then developed a deep learning-based trajectory forecast model using the collected dataset. This model can be used to predict the future trajectories of pedestrians and other vehicles in the traffic scene, which can be used by autonomous vehicles to make informed decisions about their own trajectories.

7.2 Limitations

- **Real-World Validation**: The algorithms developed in this thesis were tested in controlled simulation environments. However, their deployment and performance in real-world scenarios remain untested. Determining how these algorithms can be effectively transferred from simulation to real-world applications is an important research area that falls outside the scope of this thesis.

- **Safety Concerns**: The thesis does not fully address whether safety can be assured using solely a learning-based approach. While collision rates and success rates were analyzed in simulation, it is assumed that real-world applications will require an additional safety layer to oversee decisions made by reinforcement learning (RL) agents. The thesis demonstrates that the ego-vehicle can achieve collision-free navigation with a single pedestrian, but fails to maintain this level of safety in scenarios involving multiple vehicles.

- **State Representation Assumptions**: The deep reinforcement learning (DRL) framework utilized in this work assumes that the state of the environment is fully known to the agent. However, translating raw sensor data (such as LiDAR and camera data) into high-level state representations for autonomous vehicles (AVs) is not addressed. It is presumed that such a representation is already available to the decision-making module.

- **User Experience and Ergonomics**: Although the thesis aims to create vehicles that mimic human driving behavior, it does not account for user experience and ergonomics. Factors such as passenger comfort, perception, and interaction with AVs have not been analyzed, which could influence the overall acceptance and usability of such systems.

- **Limited Sensor Data in VR**: The VR environment developed in Chapter 6 primarily focuses on 2D pedestrian-car trajectories, without considering more complex data such as camera input or pedestrian pose. This limitation restricts the study's ability to accurately predict pedestrian behavior and represent real-world scenarios comprehensively.

- **Single Participant Scenario**: The current virtual environment is designed for interactions involving a single driver and a single pedestrian. This setup limits

the scope of the study, as it does not capture the complexities of multi-agent interactions typical in real-world traffic scenarios.

- **Data Distribution Matching**: The thesis does not adequately address the challenge of aligning data distributions between VR simulations and real-world data. Factors such as graphical fidelity, physics, and interaction dynamics in VR may differ from real-world conditions, potentially affecting the validity of the research findings.

7.3 Future Work

- **Real-World Deployment and Testing**: Future research should focus on the deployment and testing of the developed algorithms in real-world environments. This includes studying how the algorithms handle the complexities and uncertainties of real-world scenarios, beyond the controlled conditions of a simulation
- **Safety Assurance with RL**: Investigating methods to ensure safety in autonomous vehicles that rely on RL-based decision-making will be crucial. Future work should explore integrating advanced safety layers or mechanisms to monitor and control the RL agent's actions in real time, especially in scenarios involving multiple vehicles and diverse road users.
- **Enhanced State Representation from Raw Sensor Data**: Further research should explore techniques for autonomous vehicles to derive high-level state representations from raw sensor data, such as LiDAR and camera feeds. This could involve developing algorithms for sensor fusion, perception, and scene understanding to improve decision-making processes.
- **Improving User Experience and Ergonomics**: Future work could involve studying how passengers perceive and interact with autonomous vehicles and identifying ways to enhance their experience. This may include optimizing vehicle behavior for smoother rides, designing human-centric interfaces, and evaluating passenger comfort and safety.
- **Incorporation of Multi-Sensor Data in VR**: Expanding the VR environment to include multi-sensor data, such as camera input and pedestrian pose, could significantly enhance the accuracy of pedestrian behavior prediction models. Future work could focus on integrating these additional data types to make the simulations more representative of real-world conditions.
- **Multi-Agent Scenarios in VR**: Extending the VR environment to support scenarios involving multiple drivers and pedestrians would be a valuable area of future research. Such an expansion could provide deeper insights into the complex interactions between various agents and improve the robustness of autonomous vehicle models.

- **Improving Realism in VR Simulations**: Future work should aim to reduce the gap between VR and real-world conditions by improving the realism of VR simulations. This could involve enhancing graphical fidelity, refining the physics engine, and creating more sophisticated interaction dynamics to better match real-world data distributions and improve the validity of VR-based research.

Bibliography

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[2] W. H. Organization *et al.*, "Global status report on road safety 2018: Summary," World Health Organization, Tech. Rep., 2018.

[3] S. O.-R. A. V. S. Committee *et al.*, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," *SAE Standard J*, vol. 3016, pp. 1–16, 2014.

[4] M. Maurer, J. Christian Gerdes, B. Lenz, and H. Winner, *Autonomous driving: technical, legal and social aspects*. Springer Nature, 2016.

[5] M. Wu, N. Wang, and K. F. Yuen, "Deep versus superficial anthropomorphism: Exploring their effects on human trust in shared autonomous vehicles," *Computers in Human Behavior*, vol. 141, p. 107 614, 2023.

[6] S. D. Pendleton, H. Andersen, X. Du, *et al.*, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, 2017.

[7] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 187–210, 2018.

[8] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *2017 IEEE intelligent vehicles symposium (IV)*, IEEE, 2017, pp. 1671–1678.

[9] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019.

[10] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for cars that coordinate with people: Leveraging effects on human actions for planning and active information gathering over human internal state," *Autonomous Robots*, vol. 42, no. 7, pp. 1405–1426, 2018.

[11] N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, "Game theoretic modeling of vehicle interactions at unsignalized intersections and application to autonomous vehicle control," in *2018 Annual American Control Conference (ACC)*, IEEE, 2018, pp. 3215–3220.

[12] J. Li, L. Yao, X. Xu, B. Cheng, and J. Ren, "Deep reinforcement learning for pedestrian collision avoidance and human-machine cooperative driving," *Information Sciences*, vol. 532, pp. 110–124, 2020.

[13] H. Chae, C. M. Kang, B. Kim, J. Kim, C. C. Chung, and J. W. Choi, "Autonomous braking system via deep reinforcement learning," in *2017 IEEE 20th International conference on intelligent transportation systems (ITSC)*, IEEE, 2017, pp. 1–6.

[14] S. Feng, Z. Song, Z. Li, Y. Zhang, and L. Li, "Robust platoon control in mixed traffic flow based on tube model predictive control," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 711–722, 2021.

[15] A. Rasouli and J. K. Tsotsos, "Autonomous vehicles that interact with pedestrians: A survey of theory and practice," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 3, pp. 900–918, 2019.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[18] D. Silver, J. Schrittwieser, K. Simonyan, *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[19] T. P. Lillicrap, J. J. Hunt, A. Pritzel, *et al.*, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[20] X. Lei, Z. Zhang, and P. Dong, "Dynamic path planning of unknown environment based on deep reinforcement learning," *Journal of Robotics*, vol. 2018, 2018.

[21] J. Xin, H. Zhao, D. Liu, and M. Li, "Application of deep reinforcement learning in mobile robot path planning," in *2017 Chinese Automation Congress (CAC)*, IEEE, 2017, pp. 7112–7116.

[22] J. Janai, F. Güney, A. Behl, A. Geiger, *et al.*, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Foundations and Trends® in Computer Graphics and Vision*, vol. 12, no. 1–3, pp. 1–308, 2020.

[23] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[24] D. Yang, Ü. Özgüner, and K. Redmill, "A social force based pedestrian motion model considering multi-pedestrian interaction with a vehicle," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 6, no. 2, pp. 1–27, 2020.

[25] G. Markkula, R. Romano, R. Madigan, C. W. Fox, O. T. Giles, and N. Merat, "Models of human decision-making as tools for estimating and optimizing impacts of vehicle automation," *Transportation research record*, vol. 2672, no. 37, pp. 153–163, 2018.

[26] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[27] Z. Cao, E. Biyik, W. Z. Wang, *et al.*, "Reinforcement learning based control of imitative policies for near-accident driving," in *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[28] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[30] C. G. McClintock and S. T. Allison, "Social value orientation and helping behavior 1," *Journal of Applied Social Psychology*, vol. 19, no. 4, pp. 353–362, 1989.

[31] P. A. Van Lange, E. De Bruin, W. Otten, and J. A. Joireman, "Development of prosocial, individualistic, and competitive orientations: Theory and preliminary evidence.," *Journal of personality and social psychology*, vol. 73, no. 4, p. 733, 1997.

[32] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24 972–24 978, 2019.

[33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[35] G. Markkula, R. Madigan, D. Nathanael, *et al.*, "Defining interactions: A conceptual framework for understanding interactive behaviour in human and automated road traffic," *Theoretical Issues in Ergonomics Science*, vol. 21, no. 6, pp. 728–752, 2020.

[36] A. Turnwald and D. Wollherr, "Human-like motion planning based on game theoretic decision making," *International Journal of Social Robotics*, vol. 11, no. 1, pp. 151–170, 2019.

[37] L. Liu, S. Lu, R. Zhong, *et al.*, "Computing systems for autonomous driving: State of the art and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6469–6486, 2020.

[38] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus, "Intention-aware motion planning," in *Algorithmic foundations of robotics X*, Springer, 2013, pp. 475–491.

[39] A. Millard-Ball, "Pedestrians, autonomous vehicles, and cities," *Journal of planning education and research*, vol. 38, no. 1, pp. 6–12, 2018.

[40] G. Markkula, Y.-S. Lin, A. R. Srinivasan, *et al.*, "Explaining human interactions on the road by large-scale integration of computational psychological theory," *PNAS nexus*, vol. 2, no. 6, pgad163, 2023.

[41] N. AbuAli and H. Abou-Zeid, "Driver behavior modeling: Developments and future directions," *International journal of vehicular technology*, vol. 2016, 2016.

[42] N. M. Negash and J. Yang, "Driver behavior modeling towards autonomous vehicles: Comprehensive review," *IEEE Access*, 2023.

[43] S. Kolekar, J. de Winter, and D. Abbink, "Human-like driving behaviour emerges from a risk-based driver model," *Nature communications*, vol. 11, no. 1, p. 4850, 2020.

[44] J. E. Domeyer, J. D. Lee, H. Toyoda, B. Mehler, and B. Reimer, "Driver-pedestrian perceptual models demonstrate coupling: Implications for vehicle automation," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 4, pp. 557–566, 2022.

[45] D. Zhou, H. Liu, H. Ma, X. Wang, X. Zhang, and Y. Dong, "Driving behavior prediction considering cognitive prior and driving context," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2669–2678, 2020.

[46] J. Petit, C. Charron, and F. Mars, "Risk assessment by a passenger of an autonomous vehicle among pedestrians: Relationship between subjective and physiological measures," *Frontiers in neuroergonomics*, vol. 2, p. 682 119, 2021.

[47] L. Sun, W. Zhan, Y. Hu, and M. Tomizuka, "Interpretable modelling of driving behaviors in interactive driving scenarios based on cumulative prospect theory," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 4329–4335.

[48] A. Varhelyi, "Drivers' speed behaviour at a zebra crossing: A case study," *Accident Analysis & Prevention*, vol. 30, no. 6, pp. 731–743, 1998.

[49] G. Markkula, Y.-S. Lin, A. R. Srinivasan, *et al.*, "Explaining human interactions on the road by large-scale integration of computational psychological theory," *PNAS nexus*, vol. 2, no. 6, pgad163, 2023.

[50] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, "Driver behavior detection and classification using deep convolutional neural networks," *Expert Systems with Applications*, vol. 149, p. 113 240, 2020.

[51] A. Rasch, G. Panero, C.-N. Boda, and M. Dozza, "How do drivers overtake pedestrians? evidence from field test and naturalistic driving data," *Accident Analysis & Prevention*, vol. 139, p. 105 494, 2020.

[52] S. Sun, Z. Zhang, Z. Zhang, P. Deng, K. Tian, and C. Wei, "How do human-driven vehicles avoid pedestrians in interactive environments? a naturalistic driving study," *Sensors*, vol. 22, no. 20, p. 7860, 2022.

[53] P. Nasernejad, T. Sayed, and R. Alsaleh, "Multiagent modeling of pedestrian-vehicle conflicts using adversarial inverse reinforcement learning," *Transportmetrica A: Transport Science*, pp. 1–35, 2022.

[54] S. El Hamdani, N. Benamar, and M. Younis, "Pedestrian support in intelligent transportation systems: Challenges, solutions and open issues," *Transportation research part C: emerging technologies*, vol. 121, p. 102 856, 2020.

[55] R. L. Moore, "Pedestrian choice and judgment," *Journal of the Operational Research Society*, vol. 4, no. 1, pp. 3–10, 1953.

[56] D. Dey and J. Terken, "Pedestrian interaction with vehicles: Roles of explicit and implicit communication," in *Proceedings of the 9th international conference on automotive user interfaces and interactive vehicular applications*, 2017, pp. 109–113.

[57] W. Wang, L. Wang, C. Zhang, C. Liu, L. Sun, *et al.*, "Social interactions for autonomous driving: A review and perspectives," *Foundations and Trends® in Robotics*, vol. 10, no. 3-4, pp. 198–376, 2022.

[58] K. Tian, G. Markkula, C. Wei, *et al.*, "Explaining unsafe pedestrian road crossing behaviours using a psychophysics-based gap acceptance model," *Safety Science*, vol. 154, p. 105 837, 2022.

[59] D. Dey, A. Matviienko, M. Berger, B. Pfleging, M. Martens, and J. Terken, "Communicating the intention of an automated vehicle to pedestrians: The contributions of ehmi and vehicle behavior," *it-Information Technology*, vol. 63, no. 2, pp. 123–141, 2021.

[60] C. Creech, D. Tilbury, J. Yang, A. Pradhan, K. Tsui, L. Robert, *et al.*, "Pedestrian trust in automated vehicles: Role of traffic signal and av driving behavior," *Frontiers in Robotics and AI, Forthcoming*, 2019.

[61] J. Zhao, J. O. Malenje, Y. Tang, and Y. Han, "Gap acceptance probability model for pedestrians at unsignalized mid-block crosswalks based on logistic regression," *Accident Analysis & Prevention*, vol. 129, pp. 76–83, 2019.

[62] K. Tian, G. Markkula, C. Wei, *et al.*, "Deconstructing pedestrian crossing decision-making in interactions with continuous traffic: An anthropomorphic model," *arXiv preprint arXiv:2301.10419*, 2023.

[63] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Understanding pedestrian behavior in complex traffic scenes," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 61–70, 2017.

[64] K. De Clercq, A. Dietrich, J. P. Núñez Velasco, J. De Winter, and R. Happee, "External human-machine interfaces on automated vehicles: Effects on pedestrian crossing decisions," *Human factors*, vol. 61, no. 8, pp. 1353–1370, 2019.

[65] L. Kooijman, R. Happee, and J. C. de Winter, "How do ehmis affect pedestrians' crossing behavior? a study using a head-mounted display combined with a motion suit," *Information*, vol. 10, no. 12, p. 386, 2019.

[66] D. Moore, R. Currano, G. E. Strack, and D. Sirkin, "The case for implicit external human-machine interfaces for autonomous vehicles," in *Proceedings of the 11th international conference on automotive user interfaces and interactive vehicular applications*, 2019, pp. 295–307.

[67] V. Onkhar, P. Bazilinskyy, D. Dodou, and J. De Winter, "The effect of drivers' eye contact on pedestrians' perceived safety," *Transportation research part F: traffic psychology and behaviour*, vol. 84, pp. 194–210, 2022.

[68] H. Furuya, K. Kim, G. Bruder, P. J. Wisniewski, and G. F. Welch, "Autonomous vehicle visual embodiment for pedestrian interactions in crossing scenarios: Virtual drivers in avs for pedestrian crossing," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–7.

[69] P. Sewalkar and J. Seitz, "Vehicle-to-pedestrian communication for vulnerable road users: Survey, design considerations, and challenges," *Sensors*, vol. 19, no. 2, p. 358, 2019.

[70] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," in *Situational awareness*, Routledge, 2017, pp. 9–42.

[71] K. Tian, "Psychological mechanisms in pedestrian road crossing behaviour: Observations and models," Ph.D. dissertation, University of Leeds, 2023.

[72] M. Prédhumeau, L. Mancheva, J. Dugdale, and A. Spalanzani, "Agent-based modeling for predicting pedestrian trajectories around an autonomous vehicle," *Journal of Artificial Intelligence Research*, vol. 73, pp. 1385–1433, 2022.

[73] B. R. Kadali, N. Rathi, and V. Perumal, "Evaluation of pedestrian mid-block road crossing behaviour using artificial neural network," *Journal of traffic and transportation engineering (English edition)*, vol. 1, no. 2, pp. 111–119, 2014.

[74] H. Zhu, W. Alhajyaseen, M. Iryo-Asano, H. Nakamura, and C. Dias, "Defensive or competitive autonomous vehicles: Which one interacts safely and efficiently with pedestrians?" *Physica A: Statistical Mechanics and its Applications*, vol. 606, p. 128 083, 2022.

[75] J. J. Gibson, *The ecological approach to visual perception: classic edition*. Psychology press, 2014.

[76] E. R. Hoffmann and R. G. Mortimer, "Drivers' estimates of time to collision," *Accident Analysis & Prevention*, vol. 26, no. 4, pp. 511–520, 1994.

[77] P. R. DeLucia, "Critical roles for distance, task, and motion in space perception: Initial conceptual framework and practical implications," *Human Factors*, vol. 50, no. 5, pp. 811–820, 2008.

[78] B. G. Bardy and W. H. Warren Jr, "Visual control of braking in goal-directed action and sport," *Journal of Sports Sciences*, vol. 15, no. 6, pp. 607–620, 1997.

[79] D. Oberfeld, M. Wessels, and D. Büttner, "Overestimated time-to-collision for quiet vehicles: Evidence from a study using a novel audiovisual virtual-reality system for traffic scenarios," *Accident Analysis & Prevention*, vol. 175, p. 106 778, 2022.

[80] J. P. Wann, D. R. Poulter, and C. Purcell, "Reduced sensitivity to visual looming inflates the risk posed by speeding vehicles when children try to cross the road," *Psychological science*, vol. 22, no. 4, pp. 429–434, 2011.

[81] K. Jiang, F. Ling, Z. Feng, *et al.*, "Effects of mobile phone distraction on pedestrians' crossing behavior and visual attention allocation at a signalized intersection: An outdoor experimental study," *Accident Analysis & Prevention*, vol. 115, pp. 170–177, 2018.

[82] A. Theofilatos, A. Ziakopoulos, O. Oviedo-Trespalacios, and A. Timmis, "To cross or not to cross? review and meta-analysis of pedestrian gap acceptance decisions at midblock street crossings," *Journal of Transport & Health*, vol. 22, p. 101 108, 2021.

[83] V. Himanen and R. Kulmala, "An application of logit models in analysing the behaviour of pedestrians and car drivers on pedestrian crossings," *Accident Analysis & Prevention*, vol. 20, no. 3, pp. 187–197, 1988.

[84] Y. M. Lee, R. Madigan, C. Uzondu, *et al.*, "Learning to interpret novel ehmi: The effect of vehicle kinematics and ehmi familiarity on pedestrian'crossing behavior," *Journal of safety research*, vol. 80, pp. 270–280, 2022.

[85] J. Pekkanen, O. T. Giles, Y. M. Lee, *et al.*, "Variable-drift diffusion models of pedestrian road-crossing decisions," *Computational Brain & Behavior*, vol. 5, no. 1, pp. 60–80, 2022.

[86] W. Wu, R. Chen, H. Jia, Y. Li, and Z. Liang, "Game theory modeling for vehicle–pedestrian interactions and simulation based on cellular automata," *International Journal of Modern Physics C*, vol. 30, no. 04, p. 1 950 025, 2019.

[87] J. A. Oxley, E. Ihsen, B. N. Fildes, J. L. Charlton, and R. H. Day, "Crossing roads safely: An experimental study of age differences in gap selection by pedestrians," *Accident Analysis & Prevention*, vol. 37, no. 5, pp. 962–971, 2005.

[88] A. N. Tump, T. J. Pleskac, and R. H. Kurvers, "Wise or mad crowds? the cognitive mechanisms underlying information cascades," *Science Advances*, vol. 6, no. 29, eabb0266, 2020.

[89] O. Giles, G. Markkula, J. Pekkanen, *et al.*, "At the zebra crossing: Modelling complex decision processes with variable-drift diffusion models," in *Proceedings of the 41st annual meeting of the cognitive science society*, Cognitive Science Society, 2019, pp. 366–372.

[90] Y. Wang, A. R. Srinivasan, J. P. Jokinen, A. Oulasvirta, and G. Markkula, "Modeling human road crossing decisions as reward maximization with visual perception limitations," *arXiv preprint arXiv:2301.11737*, 2023.

[91] M. S. Raff *et al.*, "A volume warrant for urban stop signs," 1950.

[92] H. C. Manual, "Highway capacity manual 2010," *Transportation Research Board, National Research Council, Washington, DC*, vol. 1207, 2010.

[93] D. S. Pawar and G. R. Patil, "Critical gap estimation for pedestrians at uncontrolled mid-block crossings on high-speed arterials," *Safety science*, vol. 86, pp. 295–303, 2016.

[94] A. Rasouli and I. Kotseruba, "Intend-wait-cross: Towards modeling realistic pedestrian crossing behavior," *arXiv preprint arXiv:2203.07324*, 2022.

[95] A. H. Kalantari, Y. Yang, N. Merat, and G. Markkula, "Modelling vehicle-pedestrian interactions at unsignalised locations: Road users may not play the nash equilibrium," 2023.

[96] F. Camara, S. Cosar, N. Bellotto, N. Merat, and C. W. Fox, "Continuous game theory pedestrian modelling method for autonomous vehicles," in *Human Factors in Intelligent Vehicles*, River Publishers, 2022, pp. 1–20.

[97] W. Zeng, P. Chen, H. Nakamura, and M. Iryo-Asano, "Application of social force model to pedestrian behavior analysis at signalized crosswalk," *Transportation research part C: emerging technologies*, vol. 40, pp. 143–159, 2014.

[98] Y. Ma, E. W. M. Lee, and R. K. K. Yuen, "An artificial intelligence-based approach for simulating pedestrian movement," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3159–3170, 2016.

[99] M. Layegh, B. Mirbaha, and A. A. Rassafi, "Modeling the pedestrian behavior at conflicts with vehicles in multi-lane roundabouts (a cellular automata approach)," *Physica A: Statistical Mechanics and its Applications*, vol. 556, p. 124 843, 2020.

[100] L. Lu, G. Ren, W. Wang, C.-Y. Chan, and J. Wang, "A cellular automaton simulation model for pedestrian and vehicle interaction behaviors at unsignalized mid-block crosswalks," *Accident Analysis & Prevention*, vol. 95, pp. 425–437, 2016.

[101] A. Kalatian and B. Farooq, "A context-aware pedestrian trajectory prediction framework for automated vehicles," *Transportation research part C: emerging technologies*, vol. 134, p. 103 453, 2022.

[102] X. Zhang, H. Chen, W. Yang, W. Jin, and W. Zhu, "Pedestrian path prediction for autonomous driving at un-signalized crosswalk using w/cdm and msfm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3025–3037, 2020.

[103] J. Domeyer, A. Dinparastdjadid, J. D. Lee, G. Douglas, A. Alsaid, and M. Price, "Proxemics and kinesics in automated vehicle–pedestrian communication: Representing ethnographic observations," *Transportation research record*, vol. 2673, no. 10, pp. 70–81, 2019.

[104] R. Lobjois and V. Cavallo, "Age-related differences in street-crossing decisions: The effects of vehicle speed and time constraints on gap selection in an estimation task," *Accident analysis & prevention*, vol. 39, no. 5, pp. 934–943, 2007.

[105] G. Markkula, Z. Uludağ, R. M. Wilkie, and J. Billington, "Accumulation of continuously time-varying sensory evidence constrains neural and behavioral responses in human collision threat detection," *PLoS Computational Biology*, vol. 17, no. 7, e1009096, 2021.

[106] R. Lobjois and V. Cavallo, "The effects of aging on street-crossing behavior: From estimation to actual crossing," *Accident Analysis & Prevention*, vol. 41, no. 2, pp. 259–267, 2009.

[107] R. Lobjois, N. Benguigui, and V. Cavallo, "The effects of age and traffic density on street-crossing behavior," *Accident Analysis & Prevention*, vol. 53, pp. 166–175, 2013.

[108] K. Tian, G. Markkula, C. Wei, *et al.*, "Impacts of visual and cognitive distractions and time pressure on pedestrian crossing behaviour: A simulator study," *Accident Analysis & Prevention*, vol. 174, p. 106 770, 2022.

[109] R. Anders, F. Alario, L. Van Maanen, *et al.*, "The shifted wald distribution for response time data analysis.," *Psychological methods*, vol. 21, no. 3, p. 309, 2016.

[110] K. Tian, G. Markkula, C. Wei, and R. Romano, "Decision model for pedestrian interacting with traffic at uncontrolled intersections," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2022, pp. 183–188.

[111] B. Baker, I. Kanitscheider, T. Markov, *et al.*, "Emergent tool use from multi-agent autocurricula," *arXiv preprint arXiv:1909.07528*, 2019.

[112] K. Tian, A. Tzigieras, C. Wei, *et al.*, "Deceleration parameters as implicit communication signals for pedestrians' crossing decisions and estimations of automated vehicle behaviour," *Accident Analysis & Prevention*, vol. 190, p. 107 173, 2023.

[113] C. Zhang, B. Zhou, T. Z. Qiu, and S. Liu, "Pedestrian crossing behaviors at uncontrolled multi-lane mid-block crosswalks in developing world," *Journal of safety research*, vol. 64, pp. 145–154, 2018.

[114] J. Montufar, J. Arango, M. Porter, and S. Nakagawa, "Pedestrians' normal walking speed and speed when crossing a street," *Transportation research record*, vol. 2002, no. 1, pp. 90–97, 2007.

[115] A. Forde and J. Daniel, "Pedestrian walking speed at un-signalized midblock crosswalk and its impact on urban street segment performance," *Journal of traffic and transportation engineering (English edition)*, vol. 8, no. 1, pp. 57–69, 2021.

[116] T. Toffoli and N. Margolus, *Cellular automata machines: a new environment for modeling*. MIT press, 1987.

[117] M. Moussaid, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PloS one*, vol. 5, no. 4, e10047, 2010.

[118] X. Song, D. Han, J. Sun, and Z. Zhang, "A data-driven neural network approach to simulate pedestrian movement," *Physica A: Statistical Mechanics and Its Applications*, vol. 509, pp. 827–844, 2018.

[119] F. Martinez-Gil, M. Lozano, and F. Fernández, "Marl-ped: A multi-agent reinforcement learning based framework to simulate pedestrian groups," *Simulation Modelling Practice and Theory*, vol. 47, pp. 259–275, 2014.

[120] K. Li, S. Eiffert, M. Shan, F. Gomez-Donoso, S. Worrall, and E. Nebot, "Attentional-gcnn: Adaptive pedestrian trajectory prediction towards generic autonomous vehicle use cases," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 14 241–14 247.

[121] B. Völz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto, "A data-driven approach for pedestrian intention estimation," in *2016 ieee 19th international conference on intelligent transportation systems (itsc)*, IEEE, 2016, pp. 2607–2612.

[122] Y. Wang, H. Huang, B. Zhang, and J. Wang, "A differentiated decision-making algorithm for automated vehicles based on pedestrian feature estimation," *IET Intelligent Transport Systems*, 2023.

[123] Y. Chen, S. Li, X. Tang, K. Yang, D. Cao, and X. Lin, "Interaction-aware decision making for autonomous vehicles," *IEEE Transactions on Transportation Electrification*, 2023.

[124] A. Li, H. Jiang, J. Zhou, and X. Zhou, "Learning human-like trajectory planning on urban two-lane curved roads from experienced drivers," *IEEE Access*, vol. 7, pp. 65 828–65 838, 2019.

[125] J. Soshiroda, J. Lee, T. Daimon, and S. Kitazaki, "Stopping position matters: Drawing a better communication between pedestrian and driverless automated vehicles on narrow roads," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, SAGE Publications Sage CA: Los Angeles, CA, vol. 65, 2021, pp. 1531–1535.

[126] M. R. Bachute and J. M. Subhedar, "Autonomous driving architectures: Insights of machine learning and deep learning algorithms," *Machine Learning with Applications*, vol. 6, p. 100 164, 2021.

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

[128] Y. Xu, Z. Piao, and S. Gao, "Encoding crowd interaction with deep neural network for pedestrian trajectory prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5275–5284.

[129] H. Bi, Z. Fang, T. Mao, Z. Wang, and Z. Deng, "Joint prediction for kinematic trajectories in vehicle-pedestrian-mixed scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10 383–10 392.

[130] J. Amirian, J.-B. Hayet, and J. Pettré, "Social ways: Learning multi-modal distributions of pedestrian trajectories with gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[131] T.-Y. Chen, H.-J. Lin, C.-S. Shih, K.-T. Kuo, Q. Liu, and R. H. Chan, "Prediction of human intention in vehicles, pedestrians and bicyclists interactions," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 64–69.

[132] Y. Yoon and K. Yi, "Design of longitudinal control for autonomous vehicles based on interactive intention inference of surrounding vehicle behavior using long short-term memory," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 196–203.

[133] B. Rainbow, Q. Men, and H. P. H. Shum, "Semantics-stgcnn: A semantics-guided spatial-temporal graph convolutional network for multi-class trajectory prediction," in *Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics*, ser. SMC '21, Melbourne, Australia: IEEE, 2021, pp. 2959–2966. DOI: 10.1109/SMC52423.2021.9658781.

[134] R. Li, S. Katsigiannis, and H. P. H. Shum, "Multiclass-sgcn: Sparse graph-based trajectory prediction with agent class embedding," in *Proceedings of the 2022 IEEE International Conference on Image Processing*, ser. ICIP '22, Bordeaux, France: IEEE, 2022, pp. 2346–2350. DOI: 10.1109/ICIP46576.2022.9897644.

[135] L. Li, J. Gan, X. Ji, X. Qu, and B. Ran, "Dynamic driving risk potential field model under the connected and automated vehicles environment and its application in car-following modeling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 122–141, 2020.

[136] L. Wang, L. Sun, M. Tomizuka, and W. Zhan, "Socially-compatible behavior design of autonomous vehicles with verification on real human data," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3421–3428, 2021.

[137] X. Zhao, Y. Tian, and J. Sun, "Yield or rush? social-preference-aware driving interaction modeling using game-theoretic framework," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 453–459.

[138] G. M. J.-B. C. Chaslot, "Monte-carlo tree search," 2010.

[139] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[140] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 2111–2117.

[141] R. Chandra, A. Bera, and D. Manocha, "Using graph-theoretic machine learning to predict human driver behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2572–2585, 2021.

[142] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 9961–9980, 2021.

[143] Z. Guo, Y. Huang, X. Hu, H. Wei, and B. Zhao, "A survey on deep learning based approaches for scene understanding in autonomous driving," *Electronics*, vol. 10, no. 4, p. 471, 2021.

[144] Y. Hu, H. P. H. Shum, and E. S. L. Ho, "Multi-task deep learning with optical flow features for self-driving cars," *IET Intelligent Transport Systems*, vol. 14, no. 13, pp. 1845–1854, 2020, ISSN: 1751-956X. DOI: 10.1049/iet-its.2020.0439.

[145] S. Teng, L. Chen, Y. Ai, Y. Zhou, Z. Xuanyuan, and X. Hu, "Hierarchical interpretable imitation learning for end-to-end autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 673–683, 2022.

[146] C. Dong, J. M. Dolan, and B. Litkouhi, "Intention estimation for ramp merging control in autonomous driving," in *2017 IEEE intelligent vehicles symposium (IV)*, IEEE, 2017, pp. 1584–1589.

[147] L. Sun, C. Peng, W. Zhan, and M. Tomizuka, "A fast integrated planning and control framework for autonomous driving via imitation learning," in *Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, vol. 51913, 2018, V003T37A012.

[148] P. Cai, H. Wang, Y. Sun, and M. Liu, "Dignet: Learning scalable self-driving policies for generic traffic scenarios with graph neural networks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 8979–8984.

[149] E. Bronstein, M. Palatucci, D. Notz, *et al.*, "Hierarchical model-based imitation learning for planning in autonomous driving," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 8652–8659.

[150] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[151] Y. Zhu, D. Ren, D. Qian, M. Fan, X. Li, and H. Xia, "Star topology based interaction for robust trajectory forecasting in dynamic scene," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 3255–3261.

[152] Z. Chang, G. A. Koulieris, and H. P. H. Shum, "On the design fundamentals of diffusion models: A survey," *arXiv*, 2023. arXiv: arXiv:2306.04542 [cs.LG]. [Online]. Available: http://arxiv.org/abs/2306.04542.

[153] H. Su, J. Zhu, Y. Dong, and B. Zhang, "Forecast the plausible paths in crowd scenes.," in *IJCAI*, vol. 1, 2017, p. 2.

[154] Y. Lu, W. Wang, X. Hu, P. Xu, S. Zhou, and M. Cai, "Vehicle trajectory prediction in connected environments via heterogeneous context-aware graph convolutional networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[155] X. Li, X. Ying, and M. C. Chuah, "Grip: Graph-based interaction-aware trajectory prediction," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 3960–3966.

[156] S. Casas, C. Gulino, R. Liao, and R. Urtasun, "Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 9491–9497.

[157] X. Mo, Y. Xing, and C. Lv, "Graph and recurrent neural network-based vehicle trajectory prediction for highway driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 1934–1939.

[158] S. Chen, J. Dong, P. Ha, Y. Li, and S. Labi, "Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 7, pp. 838–857, 2021.

[159] K. Jin and X. Han, "Conquering ghosts: Relation learning for information reliability representation and end-to-end robust navigation," *arXiv preprint arXiv:2203.09952*, 2022.

[160] P. Hart and A. Knoll, "Graph neural networks and reinforcement learning for behavior generation in semantic environments," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2020, pp. 1589–1594.

[161] M. Hügle, G. Kalweit, M. Werling, and J. Boedecker, "Dynamic interaction-aware scene understanding for reinforcement learning in autonomous driving," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 4329–4335.

[162] S. Liu, K. Zheng, L. Zhao, and P. Fan, "A driving intention prediction method based on hidden markov model for autonomous driving," *Computer Communications*, vol. 157, pp. 143–149, 2020.

[163] H. Jin, C. Duan, Y. Liu, and P. Lu, "Gauss mixture hidden markov model to characterise and model discretionary lane-change behaviours for autonomous vehicles," *IET Intelligent Transport Systems*, vol. 14, no. 5, pp. 401–411, 2020.

[164] F. Garrido and P. Resende, "Review of decision-making and planning approaches in automated driving," *IEEE Access*, vol. 10, pp. 100 348–100 366, 2022.

[165] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 3389–3396.

[166] C.-J. Hoel, K. Wolff, and L. Laine, "Automated speed and lane change decision making using deep reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 2148–2155.

[167] P. Palanisamy, "Multi-agent connected autonomous driving using deep reinforcement learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–7.

[168] Y. Hu, A. Nakhaei, M. Tomizuka, and K. Fujimura, "Interaction-aware decision making with adaptive strategies under merging scenarios," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 151–158.

[169] N. Deshpande and A. Spalanzani, "Deep reinforcement learning based vehicle navigation amongst pedestrians using a grid-based state representation," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 2081–2086.

[170] D. M. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev, "Driving in dense traffic with model-free reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 5385–5392.

[171] M. Yuan, J. Shan, and K. Mi, "Deep reinforcement learning based game-theoretic decision-making for autonomous vehicles," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 818–825, 2021.

[172] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3052–3059.

[173] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 6015–6022.

[174] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[175] N. Deshpande, D. Vaufreydaz, and A. Spalanzani, "Behavioral decision-making for urban autonomous driving in the presence of pedestrians using deep recurrent q-network," in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, IEEE, 2020, pp. 428–433.

[176] H. Seong, C. Jung, S. Lee, and D. H. Shim, "Learning to drive at unsignalized intersections using attention-based deep reinforcement learning," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 559–566.

[177] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[178] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2017, pp. 23–30.

[179] Y. Shan, W. F. Lu, and C. M. Chew, "Pixel and feature level based domain adaptation for object detection in autonomous driving," *Neurocomputing*, vol. 367, pp. 31–38, 2019.

[180] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[181] M. Klimke, B. Völz, and M. Buchholz, "Cooperative behavior planning for automated driving using graph neural networks," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2022, pp. 167–174.

[182] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[183] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

[184] X. Chen and P. Chaudhari, "Midas: Multi-agent interaction-aware decision-making with adaptive strategies for urban autonomous navigation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 7980–7986.

[185] J. Li, L. Sun, J. Chen, M. Tomizuka, and W. Zhan, "A safe hierarchical planning framework for complex driving scenarios based on reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 2660–2666.

[186] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.

[187] M. Wang, S. P. Hoogendoorn, W. Daamen, B. van Arem, and R. Happee, "Game theoretic approach for predictive lane-changing and car-following control," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 73–92, 2015.

[188] G. Williams, B. Goldfain, P. Drews, J. M. Rehg, and E. A. Theodorou, "Best response model predictive control for agile interactions between autonomous ground vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 2403–2410.

[189] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games," in *2020 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2020, pp. 1475–1481.

[190] D. Fridovich-Keil, V. Rubies-Royo, and C. J. Tomlin, "An iterative quadratic method for general-sum differential games with feedback linearizable dynamics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 2216–2222.

[191] R. Spica, E. Cristofalo, Z. Wang, E. Montijano, and M. Schwager, "A real-time game theoretic planner for autonomous two-player drone racing," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1389–1403, 2020.

[192] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game-theoretic planning for self-driving cars in multivehicle competitive scenarios," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1313–1325, 2021.

[193] W. Schwarting, A. Pierson, S. Karaman, and D. Rus, "Stochastic dynamic games in belief space," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2157–2172, 2021.

[194] G. Ding, S. Aghli, C. Heckman, and L. Chen, "Game-theoretic cooperative lane changing using data-driven models," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3640–3647.

[195] A. Liniger and J. Lygeros, "A noncooperative game approach to autonomous racing," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 884–897, 2019.

[196] P. Hang, C. Lv, C. Huang, J. Cai, Z. Hu, and Y. Xing, "An integrated framework of decision making and motion planning for autonomous vehicles considering social behaviors," *IEEE transactions on vehicular technology*, vol. 69, no. 12, pp. 14 458–14 469, 2020.

[197] J. Yoo and R. Langari, "A stackelberg game theoretic model of lane-merging," *arXiv preprint arXiv:2003.09786*, 2020.

[198] E. Stefansson, J. F. Fisac, D. Sadigh, S. S. Sastry, and K. H. Johansson, "Human-robot interaction for truck platooning using hierarchical dynamic games," in *2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 3165–3172.

[199] H. Yu, H. E. Tseng, and R. Langari, "A human-like game theory-based controller for automatic lane changing," *Transportation Research Part C: Emerging Technologies*, vol. 88, pp. 140–158, 2018.

[200] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical game-theoretic planning for autonomous vehicles," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 9590–9596.

[201] M. Bahram, A. Lawitzky, J. Friedrichs, M. Aeberhard, and D. Wollherr, "A game-theoretic approach to replanning-aware interactive scene prediction and planning," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3981–3992, 2015.

[202] D. Isele, "Interactive decision making for autonomous vehicles in dense traffic," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 3981–3986.

[203] F. Laine, D. Fridovich-Keil, C.-Y. Chiu, and C. Tomlin, "Multi-hypothesis interactions in game-theoretic motion planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 8016–8023.

[204] Z. Wang, T. Taubner, and M. Schwager, "Multi-agent sensitivity enhanced iterative best response: A real-time game theoretic planner for drone racing in 3d environments," *Robotics and Autonomous Systems*, vol. 125, p. 103 410, 2020.

[205] S. L. Cleac'h, M. Schwager, and Z. Manchester, "Algames: A fast solver for constrained dynamic games," *arXiv preprint arXiv:1910.09713*, 2019.

[206] R. Tian, S. Li, N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, "Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts," in *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 321–326.

[207] F. Facchinei and C. Kanzow, "Generalized nash equilibrium problems," *Annals of Operations Research*, vol. 175, no. 1, pp. 177–211, 2010.

[208] R. O. Murphy, K. A. Ackermann, and M. Handgraaf, "Measuring social value orientation," *Judgment and Decision making*, vol. 6, no. 8, pp. 771–781, 2011.

[209] X. Ma, J. Li, M. J. Kochenderfer, D. Isele, and K. Fujimura, "Reinforcement learning for autonomous driving with latent state inference and spatial-temporal relationships," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 6064–6071.

[210] L. Crosato, C. Wei, E. S. Ho, and H. P. Shum, "Human-centric autonomous driving in an av-pedestrian interactive environment using svo," in *2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS)*, IEEE, 2021, pp. 1–6.

[211] L. Crosato, H. P. Shum, E. S. Ho, and C. Wei, "Interaction-aware decision-making for automated vehicles using social value orientation," *IEEE Transactions on Intelligent Vehicles*, 2022.

[212] D. Yang, F. T. Johora, K. A. Redmill, Ü. Özgüner, and J. P. Müller, "Sub-goal social force model for collective pedestrian motion under vehicle influence," *arXiv preprint arXiv:2101.03554*, 2021.

[213] F. Camara, N. Bellotto, S. Cosar, *et al.*, "Pedestrian models for autonomous driving part i: Low-level models, from sensing to tracking," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[214] F. Camara, N. Bellotto, S. Cosar, *et al.*, "Pedestrian models for autonomous driving part ii: High-level models of human behavior," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[215] B. Schroeder, N. Rouphail, K. Salamati, *et al.*, "Empirically-based performance assessment & simulation of pedestrian behavior at unsignalized crossings.," Southeastern Transportation Research, Innovation, Development and Education ..., Tech. Rep., 2014.

[216] D. Sun, S. Ukkusuri, R. F. Benekohal, and S. T. Waller, "Modeling of motorist-pedestrian interaction at uncontrolled mid-block crosswalks," in *Transportation Research Record, TRB Annual Meeting CD-ROM, Washington, DC*, 2003.

[217] K. Tian, G. Markkula, C. Wei, and R. Romano, "Creating kinematics-dependent pedestrian crossing willingness model when interacting with approaching vehicle," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2020, pp. 1–6.

[218] D. Yang, K. Redmill, and Ü. Özgüner, "A multi-state social force based framework for vehicle-pedestrian interaction in uncontrolled pedestrian crossing scenarios," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2020, pp. 1807–1812.

[219] C. Ningbo, W. Wei, Q. Zhaowei, Z. Liying, and B. Qiaowen, "Simulation of pedestrian crossing behaviors at unmarked roadways based on social force model," *Discrete Dynamics in Nature and Society*, vol. 2017, 2017.

[220] J. Chen, B. Yuan, and M. Tomizuka, "Model-free deep reinforcement learning for urban autonomous driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2019, pp. 2765–2771.

[221] G. Brockman, V. Cheung, L. Pettersson, *et al.*, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[222] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*, PMLR, 2018, pp. 1861–1870.

[223] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann, *Stable baselines3*, https://github.com/DLR-RM/stable-baselines3, 2019.

[224] L. Crosato, C. Wei, E. S. L. Ho, and H. P. H. Shum, "Human-centric autonomous driving in an av-pedestrian interactive environment using svo," in *Proceedings of the 2021 IEEE International Conference on Human-Machine Systems*, ser. ICHMS '21, Magdeburg, Germany: IEEE, Sep. 2021.

[225] F. Camara, N. Bellotto, S. Cosar, *et al.*, "Pedestrian models for autonomous driving part ii: High-level models of human behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5453–5472, 2020.

[226] A. Kesting, M. Treiber, and D. Helbing, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4585–4605, 2010.

[227] D. Gammelli, K. Yang, J. Harrison, F. Rodrigues, F. C. Pereira, and M. Pavone, "Graph neural network reinforcement learning for autonomous mobility-on-demand systems," in *2021 60th IEEE Conference on Decision and Control (CDC)*, IEEE, 2021, pp. 2996–3003.

[228] P. Onelcin and Y. Alver, "The crossing speed and safety margin of pedestrians at signalized intersections," *Transportation Research Procedia*, vol. 22, pp. 3–12, 2017.

[229] L. Levulytė, D. Baranyai, E. Sokolovskij, and Á. Török, "Pedestrians' role in road accidents," *International Journal for Traffic and Transport Engineering*, vol. 7, no. 3, pp. 328–341, 2017.

[230] S. Yao, J. Zhang, Z. Hu, Y. Wang, and X. Zhou, "Autonomous-driving vehicle test technology based on virtual reality," *The Journal of Engineering*, vol. 2018, no. 16, pp. 1768–1771, 2018.

[231] A. Riegler, A. Riener, and C. Holzmann, "A systematic review of virtual reality applications for automated driving: 2009–2020," *Frontiers in human dynamics*, vol. 3, p. 689 856, 2021.

[232] R. Bhagavathula, B. Williams, J. Owens, and R. Gibbons, "The reality of virtual reality: A comparison of pedestrian behavior in real and virtual environments," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, SAGE Publications Sage CA: Los Angeles, CA, vol. 62, 2018, pp. 2056–2060.

[233] I. T. Feldstein and G. N. Dyszak, "Road crossing decisions in real and virtual environments: A comparative study on simulator validity," *Accident Analysis & Prevention*, vol. 137, p. 105 356, 2020.

[234] D. Perez, M. Hasan, Y. Shen, and H. Yang, "Ar-ped: A framework of augmented reality enabled pedestrian-in-the-loop simulation," *Simulation Modelling Practice and Theory*, vol. 94, pp. 237–249, 2019.

[235] V. Kosaraju, A. Sadeghian, R. Martin-Martin, I. Reid, H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[236] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 424–14 432.

[237] A. H. Kalantari, Y. Yang, J. G. de Pedro, *et al.*, "Who goes first? a distributed simulator study of vehicle–pedestrian interaction," *Accident Analysis & Prevention*, vol. 186, p. 107 050, 2023.