POLITECNICO DI TORINO

Master of Science in Data Science and Engineering

Master's Degree Thesis

Policy Graphs and Theory of Mind for Explainable Autonomous Driving

Politecnico di Torino

Supervisor
Prof. Carlo Masone

Cosupervisors
Prof. Ulises Cortés García
Dr. Àtia Cortés Martínez
Ph.D. Student Víctor Giménez Ábalos

Candidate
Sara Montese

Academic year 2023-2024

Abstract

Thesis title: Policy Graphs and Theory of Mind for Explainable Autonomous Driving

Autonomous driving has made remarkable strides over the past two decades, propelled by advancements in artificial intelligence (AI). However, the opacity of decision-making processes in autonomous vehicles (AVs) has created significant barriers to societal trust and regulatory acceptance, primarily due to concerns surrounding trustworthiness, safety, and accountability. This thesis explores the application of Policy Graphs (PGs), an innovative explainable AI (XAI) technique, in autonomous driving. PGs represent an agent's policy as a directed graph with natural language descriptors, offering a human-readable explanation of the agent's behaviour. This framework is further enhanced by incorporating *Theory of Mind* concepts, enabling a deeper understanding of these systems as if they possessed beliefs, desires, and intentions. This approach allows the graph to capture what the agent does and what it desires and intends to do. Our research aims to make three primary contributions:

1. A comprehensive review of current XAI techniques in the context of autonomous driving provides a solid foundation for our work.
2. Development of an advanced framework that integrates the agent's desires and intentions into Policy Graphs, thus facilitating the extraction of motivations behind specific driving decisions and identifying abnormal or undesirable behaviours.
3. An exploration of how external factors, such as weather and lighting conditions, influence AV decision-making, uncovering potential harmful biases and patterns under various driving scenarios.

The findings of this study show that combining Policy Graphs and Theory of Mind concepts offers an effective approach to explaining and interpreting vehicle behaviour. This innovative methodology significantly enhances our understanding of autonomous driving systems. More importantly, it has the potential to increase public trust and contribute to more robust regulatory frameworks in the field of autonomous vehicles, thereby contributing to the safe and widespread adoption of AV technology.

Keywords: Explainable AI, Policy Graphs, Autonomous Driving, Theory of Mind

Acknowledgements

I would like to express my deepest gratitude to Prof. Ulises Cortés García. Your guidance throughout this project has been invaluable. Thank you for believing in me and for opening my mind to the world of ethics. I am also immensely grateful for efforts in dealing with the bureaucratic matters that made this thesis possible.

I extend my thanks to my co-supervisor, Àtia Cortés Martínez, for introducing me to this project and connecting me with our European partners. Thank you for your encouragement and support, beyond just academics.

A big thank you goes to Víctor Giménez Ábalos for his precious guidance through this thesis. Your patience and dedication in helping me shape the outcome of this research have brought my work to a higher level.

I am deeply thankful to Prof. Carlo Masone for encouraging my decision to pursue my thesis abroad, as well as for his support and motivation from Italy, especially for demonstrating his extraordinary availability every day.

I would like to extend my thanks to Vice-dean Oscar Romero Moral for handling the bureaucratic challenges to make my graduation possible.

To my family, thank you for your love and support, despite the distance; I could not have done this without you.

Contents

List of Figures

List of Tables

Acronims

AI Artificial Intelligence

DL Deep Learning

PG Policy Graph

ToM Theory of Mind

ML Machine Learning

AV Autonomous Vehicle

AD Autonomous Driving

XAI Explainable Artificial Intelligence

DRL Deep Reinforcement Learning

MCTS Monte Carlo Tree Search

CNN Convolutional Neural Network

MDP Markov Decision Process

WCC Weakly Connected Componenets

AUC Area Under the Curve

NLL Negative-Log-Likelihood

Chapter 1

Introduction

Artificial Intelligence (AI) has witnessed significant advancements across various fields in recent decades. Deep Learning (DL) has achieved performance levels that often match or surpass human capabilities in a wide range of tasks, thanks to its ability to process vast amounts of data and learn complex patterns [2]. As a result, DL has become a key technology in the field of Autonomous Driving (AD), where it is applied in numerous areas such as vehicle routing, traffic prediction and pedestrian behaviour modelling. Despite the promising potential of autonomous vehicles to revolutionise daily transportation, they raise concerns regarding their functional safety. Most DL applications are not safety-critical, meaning that system failures in these contexts generally do not lead to severe consequences. However, when DL is used to make life-critical decisions, such as those required for Autonomous Vehicles (AVs), the reliability and robustness of these systems become crucial from a safety perspective [3].

According to the American Automobile Association's annual automated vehicle survey, 91% of the US drivers interviewed either express fear or uncertainty towards self-driving vehicles [4]. This reluctance persists even though human error, often caused by distractions, impaired driving, traffic violations, or limited visibility, remains the leading cause of road accidents [5]. Reports of accidents involving AVs, frequently attributed to their opaque and unpredictable decision-making processes, contribute to these concerns. A very famous case is the Uber vehicle fatal collision with a pedestrian wheeling a bicycle across the road [6]. A promising approach to address these issues is Explainable AI, which aims to make AI systems' decision-making processes understandable to humans. Understandability in AV systems is indeed essential to a variety of stakeholders. For developers, understandability allows for verification that the system operates as intended; for regulators, it enables proper auditing and certification; and for insurance companies, it facilitates the assessment of liability in the event of accidents [2].

To design and implement models capable of offering meaningful explanations, taking inspiration from human cognitive processes can be a promising starting point [7]. Building on this foundation, this work explores an innovative XAI technique in the context of autonomous driving. Specifically, it investigates the use of Policy Graphs [8], a graph-based approach for representing the behaviour of opaque agents, and integrates concepts from the Theory of Mind to generate human-understandable explanations[1]. This research builds on previous works by Gimenez-Abalos *et al.* [9, 10, 11], which discuss *how* PGs can be enhanced by thinking of agents as if they possess beliefs, desires and intentions.

1.1 Research Aims and Objectives

This thesis is part of the EU-funded project AI4CCAM (Trustworthy AI for Connected, Cooperative Automated Mobility), which aims to build trustworthy-by-design AI solutions for autonomous mobility applications. Within this broader framework, the specific aim of the thesis is to explore the application of PGs combined with the Theory of Mind in driving scenarios. Additionally, this research aims to identify any harmful biases within the driving scene dataset. Previous research has demonstrated that Policy Graphs can be traversed to answer straightforward questions about an agent's decisions, such as situational behaviour (*"What will you do when you are in state s?"*), action conditions (*"When do you perform action a?"*), and counterfactual explanations (*"Why did you not perform action X in state y?"*) [12]. However, these explanations are limited, as they

[1]The source code of the work contained in this thesis is available on Github at https://github.com/SaraMo14/nuscenes-pg.

primarily focus on short-term behaviour and do not take into account the agent's goals, desires or intentions. Consequently, they fall short of providing more goal-oriented explanations, which are essential for a deeper understanding of the agent's long-term behaviour and reasoning. By incorporating Theory of Mind into Policy Graphs, this research aims to address long-term behavioural questions, such as *"What are the agent's intentions in this state?"*. To guide the research, the following questions have been formulated:

- What XAI techniques have been implemented in the field of autonomous driving?
- Can the combination of Policy Graphs and Theory of Mind provide teleological explanations for an AV behaviour?
- What potential harmful biases exist in autonomous driving datasets, and what patterns emerge from observed driving behaviours?

1.2 Structure of the Thesis

The structure of this work is as follows. We begin by introducing the concept of Explainable Artificial Intelligence (Section 2), where we define explanation and discuss XAI's significance. This is followed by a review of state-of-the-art techniques in XAI applied to autonomous driving systems. In Section 3, we explore the concept of harmful bias in Artificial Intelligence, identifying its sources within machine learning processes and examining its impact on autonomous driving systems. This section also discusses strategies for detecting and mitigating harmful bias. Next, we describe the dataset used in this research (Section 4) and present the theoretical framework for the proposed approach, which integrates Policy Graphs with the Theory of Mind to generate explanations (Section 5). In Section 6, we demonstrate the application of Policy Graphs and Theory of Mind to extract explanations for decision-making processes in driving scenarios. Results include the analysis of behavioural patterns in driving alongside a quantitative evaluation of the dataset with a focus on identifying biases (Section 7). Finally, the thesis concludes by summarising the key findings, discussing the limitations of the proposed approach and offering suggestions for future research directions (Section 8).

Chapter 2

Explanations in Autonomous Driving

Understanding the rationale behind a model's decisions allows users to identify vulnerabilities, build trust, ensure legal compliance, and enhance performance. The collection of methods that enables human users to explain the results and output created by AI systems is referred to as Explainable Artificial Intelligence (XAI).

2.1 What Is An Explanation?

According to Encyclopædia Britannica [13], an explanation is a set of statements that clarify the existence or occurrence of an object, an event, or a state of affairs. In any explanation, two key subjects are involved: the *explainer*, the one who provides it (*e.g.* a model), and the *explainee*, the one who receives it (*e.g.* human user).

Having established the definition of an explanation, attention turns to the characteristics that constitute a *good* explanation. When an explainer provides an explanation to an explainee, they are essentially engaging in a conversation. Therefore, for an explanation to be effective, it must follow Grice's maxims of conversation [14]:

- *Quantity*: The explanation should provide the right amount of information. It should be concise but still informative.
- *Quality*: The explanation should contain reliable information, avoiding statements that are known to be false or that lack sufficient evidence (*reliability).*
- *Relation*: The explanation should be relevant, addressing the specific question of the explainee.
- *Manner*: The explanation should be understandable and not ambiguous to the recipient (*interpretability*).

2.2 Importance of XAI

In order to fully understand the importance of XAI, we first provide the definitions of the different types of models under analysis. On the one hand, there are *black-box models* (*e.g.* DL algorithms) whose internal decision-making processes are opaque. While these models can achieve exceptional performances, they lack transparency, making it difficult for users to understand how decisions are made [15]. On the other hand, *white-box models* (*e.g.* decision trees) have decision-making processes that are more straightforward to interpret, typically at the expense of performance. In addition, when differentiating between opaque and transparent models, it is important to clarify the terms *interpretability* and *explainability*, as they are often confused or used interchangeably. According to Miller [7], *interpretability* is the degree to which a human can understand the cause of a decision made by a model. Instead, we call a system *explainable* if its output is supported by an explanation [2]. It is important to note that not all black-box systems need to be explained, as it may not be necessary if the system is not safety-critical or if the problem has been sufficiently studied and validated in real-world applications, leading to a certain level of trust in the system's decisions [16].

XAI has been used in different tasks and modalities. In particular, explainability algorithms are pervasively used to [17]:

- *justify* that black-box systems operate within acceptable ethical and legal boundaries. For instance, in the case of discriminatory loan approvals, individuals may face biased decisions without a clear understanding of how the algorithm arrived at its conclusions. In such cases, the system must be able to justify why it granted or denied a loan to an individual. Furthermore, the need for AI-based systems to provide explanations aligns with regulatory requirements, such as the *Right To Explanation* under the General Data Protection Regulation (GDPR) [18] and the AI Act [19].
- *control* black-box systems for possible failures and vulnerabilities, allowing developers to identify and correct them.
- *improve* over existing models. When a system can be explained and understood, it becomes easier to enhance its performance. For instance, if a healthcare AI-based model explains that certain patient characteristics led to a specific risk prediction, clinicians can use that feedback to refine the model, leading to better predictive accuracy.
- *discover* new knowledge through explanations. For instance, if a DL model is developed to assist in diagnosing diseases based on patient symptoms and historical data, and achieves exceptionally high accuracy, it becomes crucial for the model to explain its reasoning, as it can reveal important medical knowledge hidden in the data.

This thesis focuses on justifying and uncovering driver behaviour within the dataset to support trustworthiness in autonomous vehicles. This includes identifying driving patterns or any potential harmful bias present in the driving scene dataset.

2.3 XAI Taxonomy

A variety of taxonomies has been proposed in the literature to classify different explainability methods [20]. We synthesised these existing taxonomies to create a classification that offers an adequate level of detail for our study. Fig. 2.1 illustrates the proposed taxonomy, which identifies the following main categories:

- **Scope of Explanation** (Global vs Local): Global models provide explanations of the overall behaviour of the model, while local models focus on explanations for specific decisions.
- **Stage of Explanation** (*Post-hoc* vs *Ante-hoc*): *Post-hoc* methods involve creating a surrogate model to generate explanations, whereas *ante-hoc* methods entail the direct construction of interpretable models. *Post-hoc* approaches can be further categorised based on **applicability** in *model-specific* (applicable to specific models) and *model-agnostic* (applicable to all types of models). By definition, *ante-hoc* methods are model-specific.
- **Output Format of Explanation**: The output format is the modality in which explanations are presented to stakeholders (*e.g.* visual, textual). Designing XAI models requires careful consideration of the target audience, as different stakeholders will require different types and levels of explanation.
- **Time Horizon of Explanation** (Reactive vs Proactive): Reactive models focus on providing explanations relevant to the immediate context, relying on short-term information. In contrast, proactive models focus on longer-term implications and consequences.

2.4 XAI in Autonomous Driving

AVs are sophisticated vehicles equipped with advanced sensors, such as cameras, LiDAR, GPS, RADAR, and state-of-the-art algorithms which allow them to navigate without human intervention [21]. Advancements in DL have accelerated the processing of these inputs, enabling them to analyse and solve the many tasks necessary for driving. As these vehicles are required to operate in complex, dynamic, and unpredictable environments where their decisions can have serious real-world consequences, including accidents and potential loss of life, the need for XAI is critical in AV technology. In particular, the main reasons are:

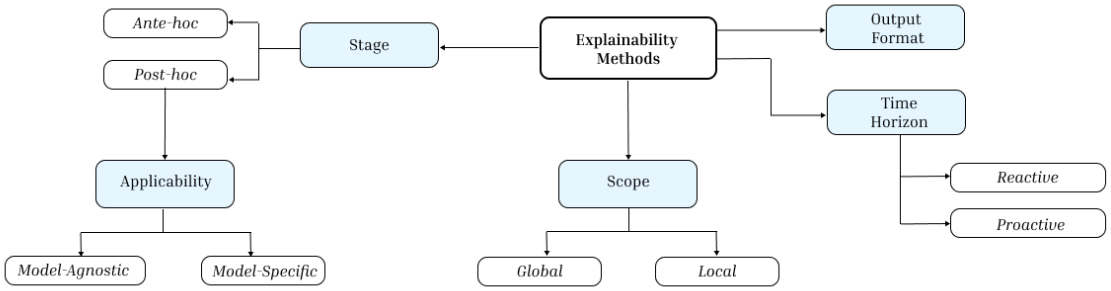


Figure 2.1: Proposed taxonomy of XAI methods

- Functional Safety: Errors or incorrect behaviours in AVs can result in serious injuries or death to individuals. Given their reliance on black-box models, AVs need to provide sufficient transparency to stakeholders (*i.e.* engineers, regulators and users) so that they can understand the rationale behind the vehicle's actions.
- Accountability and Trust: Public trust is a major obstacle to the widespread adoption of AVs. Incidents such as the fatal Uber self-driving car accident in 2018 [6] have raised serious concerns about the reliability of these systems. When AVs malfunction or exhibit unpredictable behaviour, XAI allows developers, regulators, and the public to understand the underlying causes, thus ensuring accountability.
- Ethical Decision-Making: AVs must also address ethical dilemmas, such as the Trolley problem, scenarios where the vehicle must choose between two undesirable outcomes, such as avoiding a pedestrian by crashing into another vehicle. XAI can provide transparency into how these ethical decisions are made by autonomous systems [22]. For example, if an AV chooses to prioritise the safety of pedestrians over its passenger(s) by crashing into another vehicle, XAI can clarify the factors that influenced the system's decision (e.g. the number of lives saved).
- Regulatory Compliance: The increasing presence of AVs on public roads brings with it the need for regulatory frameworks that address the transparency, safety, and ethical operation of AI-based systems. However, to date, no specific requirements for AI in autonomous driving have been established [2]. As a result, developers must rely on broader standards for trustworthy AI, such as the AI Act [19] and the General Data Protection Regulation (GDPR) [18]. The AI Act introduces strict requirements on high-risk AI systems, such as those integrated in AVs. These requirements include risk management assessments, human oversight, and transparency. In particular, the emphasis on transparency within the AI Act aligns with the GDPR's *Right To Explanation*, which mandates that users have the right to understand the rationale behind AI-driven decisions, a principle that becomes crucial in the context of AVs.

2.4.1 State Of The Art

There have been many attempts to address the aforementioned needs, and we will focus on discussing those that explain the planning behaviour of an AV. However, more comprehensive reviews exist [2, 22]. The planning behaviour of an autonomous vehicle involves making informed decisions to achieve high-level goals (*e.g.* selecting the optimal route to a destination). XAI methods to explain AV planning can be divided into three main types [22]:

- **Vision-based XAI:**
  As Convolutional Neural Networks (CNNs) enhance the visual capabilities of AVs, many techniques have been proposed to understand how these models identify image segments that influence vehicle behaviour. One prominent approach involves the generation of heat maps, which visually explain the regions in the input that the opaque algorithm focuses on when making decisions. Previous works, including Grad-CAM [23], its extensions [24], and saliency maps [25], demonstrate that heat maps can provide local, *post-hoc*, model-specific (applicable only to CNNs), reactive explanations. However, visual-based XAI have notable limitations. Ghorbani *et al.* [26] demonstrate this fragility by showing that small alterations to the input instance can result in significant changes in the generated explanations. Additionally,

heat maps can be difficult for non-expert users to interpret, complicating the assessment of the accuracy of the explanations provided. More importantly, existing vision-based XAI techniques for AV are not proactive, since they focus on immediate contexts and rely heavily on short-term information, which may not capture the rationale behind the agent's decisions.

- **Relevance-based XAI:**
  These approaches provide a quantitative measure of how much each input contributes to the opaque model's decision. A widely used method for this purpose is SHAP (SHapley Additive exPlanations) [27], which uses Shapley values to represent marginal contributions of individual features. This technique aims to quantify how each feature impacts the model's decisions. In particular, Cui *et al.* [28] combine SHAP and Random Forests to improve transparency of the decision-making process of a Deep Reinforcement Learning (DRL) algorithm in an autonomous driving scenario. In this approach, SHAP identifies key features influencing the DRL model's decisions, and a Random Forest is trained on these features to explain the original model's behaviour. This is a *post-hoc*, model-specific, reactive method that provides both global and local explanations. However, due to several limitations, relying on SHAP for interpretability may not guarantee safer autonomous driving applications. SHAP-based methods can produce inconsistent explanations compared to other feature saliency methods. Furthermore, feature attribution methods are often difficult to interpret, especially for non-expert users [2]. Additionally, perturbation-based methods like SHAP are known to be vulnerable to adversarial attacks, making them less reliable for safety-critical tasks like driving [29].

- **Textual-based XAI:**
  This category includes all methods that explain AV decisions using natural language descriptors. One promising method is the Monte Carlo Tree Search (MCTS) algorithm, which combines random sampling (Monte Carlo) with tree-based exploration to find the most promising decision [30]. Gyevnar *et al.* [31] propose a method that integrates MCTS with Bayesian Networks, a probabilistic graphical model that represents sets of random variables and their conditional dependencies through directed acyclic graphs [32]. The authors introduce an explanation generation system that maps an MCTS-based planning framework onto a Bayesian Network that models causal relationships within the planning process. This allows the retrieval of the causes and effects of an AV's actions. This method operates in an *ante-hoc* and proactive way, offering global explanations and with outputs provided in a textual format. However, this methodology heavily relies on trajectory predictions for other traffic participants, which can be computationally intensive. Another promising approach is CEMA [33], an XAI technique designed to generate causal natural language explanations for an agent's decisions in a dynamic, sequential multi-agent system, taking inspiration from social sciences. CEMA achieves this by explaining the ego's actions in terms of both the agent's teleological motivations (which describe the purpose or goal behind an action) and the influence of other agents in its environment. This approach's limitation is that it is model-specific, as it assumes that the agent operates in a dynamic sequential multi-agent system.

With the advent of Large Language Models (LLMs) and Vision-Language Models (VLMs), innovative methods are emerging for interpreting the decisions made by AVs and describing traffic scenes. However, relying on these language models for XAI introduces several challenges, including a lack of transparency and inconsistency, as LLMs can produce varying responses to similar inputs. Additionally, these models are prone to generating fictitious explanations, often referred to as "hallucinations," which can pose significant risks in driving contexts.

A comparison of the analysed literature is illustrated in Table 2.1. Having identified the limitations of these state-of-the-art methods, this thesis proposes a novel approach for explaining driving behaviour: Policy Graphs with Theory of Mind. This method addresses the aforementioned limitations by offering both global and local explanations of the agent's behaviour and operating as a *post-hoc*, model-agnostic solution, as it applies to all types of models. Furthermore, it provides proactive explanations focusing on the agent's desires and intentions rather than just the short-term context. The output is expressed in natural language predicates, ensuring it is easily interpretable for the explainee. The methodology is detailed in §5.

Table 2.1: Comparison of XAI Algorithms for Autonomous Vehicles

Algorithm	**Description**	**Scope**	**Stage**	**Time Horizon**	**Output**
Kolekar *et al.* [24]	Grad-CAM-based	Local	*Post-hoc*, Model-specific	Reactive	Visual
Mankodiya *et al.* [25]	Saliency-Map-based	Local	*Post-hoc*, Model-specific	Reactive	Visual
Cui *et al.* [28]	SHAP + Random Forests	Global/Local	*Post-hoc*, Model-specific	Reactive	Importance
Gyevnar *et al.* [31]	MCTS + Bayesian Networks	Global	*Ante-hoc*	Proactive	Textual
Gyevnar *et al.* [33]	Social sciences inspired causal explanations	Global	*Post-hoc*, Model-specific	Proactive	Textual
Policy Graphs and ToM	Intention-aware probabilistic graphical model	Global/Local	*Post-hoc*, Model-agnostic	Proactive	Textual

Chapter 3

Harmful Bias in Artificial Intelligence

AI systems are already used to drive cars, serve personalised advertisements, match individuals on dating apps, flag unusual credit-card transactions and many other tasks. Still, when performing these actions, they are not neutral. This is a primary issue, given these algorithms' impact on people's lives. In a legal sense, *Bias* is defined as *"judgement based on preconceived notions or prejudices, as opposed to the impartial evaluation of facts"* [34]. It becomes *harmful* when these prejudices in AI models lead to unfair treatment of individuals or groups, often based on attributes such as race, gender or socioeconomic status.

3.1 Sources of Bias in AI

Addressing harmful bias in AI-based systems requires an understanding of where it originates. According to Hellström *et al.* [35], sources of *bias* can be classified into three categories: bias in the world, bias in the data, and bias in the learning process. Each source can contribute to what is known as *model bias*, which refers to the bias observed in the final model outputs.

3.1.1 Bias in World

Bias in the world refers to the reflection of pre-existing inequalities and societal disproportions that are inherently embedded in data, often referred to as *historical bias* [35]. This form of bias is particularly prominent in Natural Language Processing models, where language patterns may unintentionally perpetuate and reinforce stereotypes. Lam et *al.* [36] demonstrated that when searching for "CEO" in Google Image Search, the results predominantly depicted white males, mirroring the societal imbalance in leadership roles. Similarly, the analogy "*man* is to *doctor* as *woman* is to *nurse*" is a well-known example of gender bias in word embeddings [37], reflecting historical gender roles in language data. What is quite troubling is that attempts to debias such systems may be just partial and leave intact most stereotypical associations while only addressing the most visible and measurable ones.

3.1.2 Bias in Data

AI can make biased decisions because it learns from the patterns present in the data it is trained on. If this data is biased, the resulting AI models not only inherit these biases but can also perpetuate and even amplify them in their outputs. The main types of data-related bias include [38]:

- Representation Bias, which occurs when the sample data used to train the models does not adequately reflect the diversity of the population data. This often happens when non-representative samples are collected, leading to missing subgroups or over-representation of certain groups.
- Measurement Bias, which arises when the methods or tools used to collect data are flawed, resulting in inaccuracies or skewness.
- Labelling Bias, which results from inaccuracies in labelling the training data. For instance, the human labelling process can be biased, as labelers have personal beliefs, cultural backgrounds or values that influence their judgements.

- Inherited Bias, which emerges when algorithms produce inputs for other models, perpetuating biases from the original model throughout the decision-making process.

3.1.3 Bias in Learning

Bias can also emerge during the learning process through:

- Algorithmic Bias, which arises from the design choices (such as using certain optimisation functions [38]), embedded within the algorithm itself. They are often unintentional but can significantly influence the outcomes generated by the system.
- Popularity Bias, which occurs when the system favours well-known options over less popular alternatives. For instance, in algorithmic content recommendations on social media, AI algorithms reinforce existing biases by prioritising content that aligns with user preferences.
- Evaluation Bias, which emerges when biased criteria are used to assess the performance of a model, such as when training and evaluation data differ significantly.

3.2 Impact of Bias on Autonomous Driving Systems

We aim for autonomous vehicles to be *fair*, ensuring that they treat both privileged and underprivileged groups safely and equitably. However, representation and algorithmic bias, among other biases, pose significant risks in AV technology. A study from the Georgia Institute of Technology [39] revealed alarming disparities in pedestrian detection accuracy based on skin tone, indicating that individuals with dark skin were detected with lower precision compared to light-skinned individuals. This disparity persisted even after controlling for factors such as time of day and visibility conditions.

Children pose an additional challenge for AV technology. Their smaller stature and unpredictable behaviour, combined with the fact that they can be carried in strollers or other objects, make them more difficult to detect. Given these factors, one would reasonably expect AVs to exercise greater caution around potential child encounters, yet this has not been the case. General Motors' autonomous robotaxi division, Cruise, continued to operate vehicles that struggled to recognise children adequately [40]. Internal reports indicated that the company was aware of this weakness but still proceeded with service expansions, raising severe concerns about prioritising operational goals over safety.

Another significant issue is the lack of representation of individuals with impaired mobility in the datasets used to train and evaluate AV systems. The absence of diverse data limits the ability of AVs to effectively navigate environments that include pedestrians with disabilities, potentially increasing the risk of accidents. This under-representation is also marked for those with intersecting marginalised identities, such as non-binary individuals and people of colour.

A final example of algorithmic bias in AVs is referred to as *transfer context bias* [41]. If autonomous systems are primarily trained on data collected from urban environments in the United States, they may struggle to generalise to the diverse driving conditions, road layouts, and traffic laws in other regions, such as the United Kingdom. This lack of adaptability can lead to unsafe driving decisions, as the model may have difficulty interpreting scenarios it has not been adequately trained to handle. When an AV model is designed for a specific environment, the assumptions made during the training phase can pose significant risks when applied elsewhere.

3.3 Mitigation of Bias in Autonomous Driving Systems

Achieving complete fairness is challenging, as fairness is neither binary nor absolute, and what fits one situation will not fit another [42]. However, there are several strategies that we can employ to reduce bias in autonomous driving systems:

- Inclusive Data Collection: The driving scenarios on which we build these systems should include a wide range of conditions, including urban, suburban and rural environments, as well as varying weather and light conditions. Additionally, the dataset must represent diverse demographic groups, accounting for differences in age, race, gender and abilities to ensure that vehicles can recognise and interact safely with all individuals.

- Bias Mitigation Techniques: Building an inclusive dataset may be expensive, therefore the focus should also be on bias detection and mitigation during the model development process. Techniques such as data augmentation can help balance the training dataset and reduce the influence of over-represented scenarios. Additionally, adversarial training methods can be implemented, where the model is exposed to biased inputs to learn how to correct its responses.
- Explainable AI: XAI can shed light on the decision-making process of these vehicles, helping to identify patterns in the behaviour in different scenarios and analysing which features influence decisions.
- Diversified Team: A diversified team of developers and researchers with different experiences and backgrounds increases the ability to detect and understand biases embedded in the model. The varied perspectives that come from different disciplines and life experiences enable the team to identify potential issues that may not be immediately evident to a more homogeneous group.

Chapter 4

NuScenes for Autonomous XAI

This study uses nuScenes [1], a public dataset for autonomous driving provided by Motional. The dataset includes a vast amount of sensor data collected from an autonomous vehicle navigating urban environments. The data was collected in the Boston Seaport area and several locations in Singapore, including One North, Queenstown, and Holland Village, involving both left-hand and right-hand driving. The vehicle used in the driving scenes is a Renault Zoe with a length of $4.084m$ and a width of $1.730m$.

The dataset contains 1,000 driving scenes and features data from a full suite of automotive sensors, including LiDAR, RADAR, GPS and CAN bus data. LiDAR provides high-resolution 3D maps of the vehicle's environment, enabling precise obstacle detection. RADAR uses radio waves to detect objects and measure their speed and distance, performing well in long-range detection and adverse weather conditions, complementing LiDAR. GPS provides accurate vehicle positioning for mapping and navigation. Lastly, CAN bus data records internal vehicle information such as speed, steering angle, acceleration and turn signals. The ego vehicle is further equipped with six cameras (front, back, front-left, front-right, back-left and back-right) that provide a panoramic view (360-degree) of the surroundings of the vehicle. The sensor set-up is illustrated in Figs. 4.1 and 4.2.

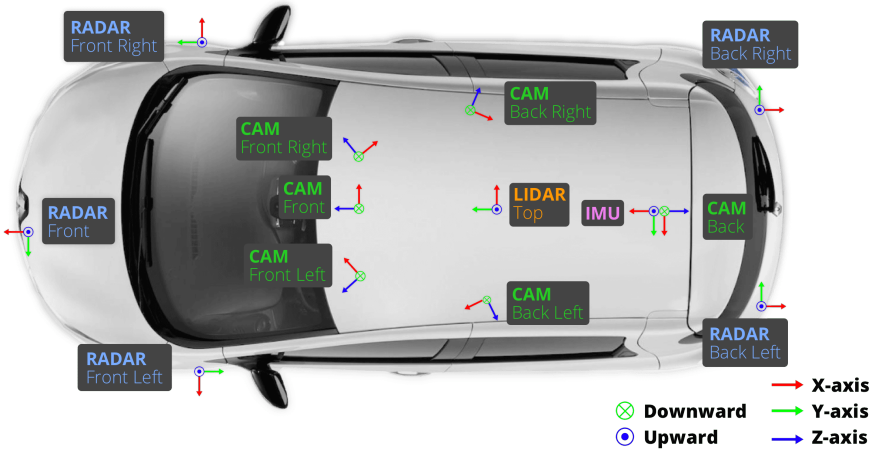


Figure 4.1: Sensor set up of the ego vehicle [1]

After collecting the driving data, synchronised keyframes consisting of images, LIDAR and RADAR data were sampled at 2Hz and annotated through experts and several validation steps. Each keyframe is annotated with 3D bounding boxes for 23 object categories (listed in Table 4.1), with each object being assigned attributes including visibility, activity, and pose. The objects are represented as cuboids and described by parameters for x, y, z coordinates, width, length, height, velocity, and yaw angle.

For this study, we focus on using the dataset to analyse the trajectories of the ego vehicle within various scenes, through a detailed preprocessing and transformation pipeline. Our analysis is limited to keyframes, thus considering camera, LiDAR and RADAR data only. Initially, the yaw angle was used to determine the vehicle's rotation and directional movement. However, discrepancies emerged between observed vehicle behaviour and corresponding yaw values in certain scene frames.

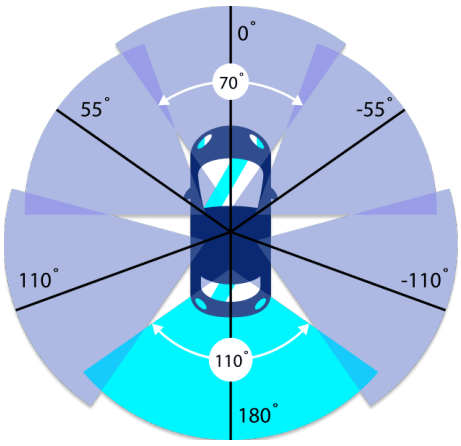


Figure 4.2: Camera set orientation and overlap [1]

The reason behind these inconsistencies is that yaw angle values are influenced by multiple factors, such as steering inputs, road conditions and external forces. This led to the integration of steering angle information from CAN bus data. Since CAN bus data and keyframes' sensor data are not synchronised, we align them based on the closest available timestamps. It is important to note that 150 out of 1,000 scenes lack annotations, as these are reserved for an ongoing public challenge. As a result, in our version of the dataset, we limit our analysis to scenes from the nuScenes training set. Furthermore, fifteen scenes from the nuScenes training did not contain CAN bus data and were excluded from the analysis. Additionally, five scenes were identified as having incorrect ego pose measurements and excluded, leaving us with 830 samples for the final analysis.

Table 4.1: Taxonomy of different object categories and subcategories in the surroundings of the ego vehicle [1]

Category	**Subcategory**	**N. Bounding Boxes**	**Attributes**
Pedestrian	Adult	208,240	Moving, Sitting or Lying Down, Standing
	Child	2,066	
	Construction Worker	9,161	
	Portable Mobility Vehicle	395	
	Police Officer	727	
	Wheelchair	503	
	Stroller	1,072	
Movable Object	Barrier	152,087	
	Debris	3,016	
	Pushable Pullable Object	24,605	
	Traffic Cone	97,959	
Static Object	Bicycle Rack	2,713	With Rider, Without Rider
Vehicle	Bicycle	11,859	With Rider, Without Rider
	Bendy Bus	1,820	Moving, Parked, Stopped
	Rigid Bus	14,501	
	Car	493,322	
	Construction Vehicle	14,671	
	Ambulance	49	
	Police Vehicle	638	
	Motorcycle	12,617	
	Trailer	24,860	
	Truck	88,519	
Animal		787	

4.1 Agent State Space

The state s of the ego vehicle at a given timestamp is extracted from the dataset. It includes several parameters:

$$s = (x, y, v, a, \psi, \delta)$$

- x and y represent the global position of the vehicle in Cartesian coordinates.
- v represents the speed of the vehicle in m/s.
- a represents the acceleration of the vehicle in m/s^2.
- ψ represents the yaw angle of the vehicle in rad/s.
- δ represents the steering angle of the vehicle in $radians$.

The yaw angle gives a global measure of the vehicle's orientation, helping to understand its overall heading direction. On the other hand, the steering angle provides a more direct mapping to the vehicle control inputs during manoeuvres. We extend the vehicle's state by incorporating information about observations from the surrounding environment. At each timestamp, we include the frequency with which objects of a certain category (*e.g.* pedestrian) and attribute (*e.g.* moving) are detected. To improve reliability of our data, we consider only those objects that exhibit a panoramic visibility greater than 60%. This visibility metric, defined as the percentage of an object's pixels visible in the combined panoramic view of all cameras, is important because highly visible objects are more likely to influence driving behaviour. While less visible objects also play a role in influencing behaviour, the 60% threshold strikes a balance by reducing the number of detections that may not significantly affect the AV decision-making process. This processing step yields a final dataset that includes both the state and observational features necessary for constructing the PG. Varying visibility thresholds for different object classes (*e.g.* lower for vulnerable road users) can be explored in future work.

4.2 Agent Action Space

Many factors, including experience, psychological conditions and individual preferences (*e.g.* aggressive versus calm driving styles), make driving behaviours varied. Ideally, an agent's range of actions would include accelerating, decelerating, turning, going in reverse, lane shifts, parking and other manoeuvres. However, complex actions have not been included in the analysis, as the dataset does not contain relevant scenes displaying these behaviours. The finite set of actions considered for this study include:

- *Gas, Brake*: Increasing or decreasing speed beyond a specified threshold while going forward (*i.e.* lane keeping).
- *Go Straight*: Maintaining a constant speed while moving forward.
- *Turn Right*, *Turn Left*: Performing a turn while keeping constant speed.
- *Gas + Turn Right*, *Gas + Turn Left*: Performing a turn while accelerating.
- *Brake + Turn Right*, *Brake + Turn Left*: Performing a turn while decelerating.
- *Idle*: The vehicle's engine is left running, but the agent is not in motion.

One potential area for improvement is the differentiation between degrees of acceleration (*e.g.* sharp versus slight) and turning (*e.g.* sharp versus gradual turns).

4.3 Agent Environment

The nuScenes Map API [1] provides detailed information about the static environment in which the ego vehicle operates. There are four different maps, one for each city in the dataset. The main elements of the road infrastructure are:

- Drivable Area: The area where the car is allowed to drive, without considering driving directions or legal restrictions.
- Stop Line: Region on the map where vehicles typically come to a halt. Reasons for a stop area include crosswalks, traffic lights, stop signs, yield signs, and the need to yield to oncoming traffic when making a simple left turn or when making a left turn and encountering a pedestrian crossing.
- Traffic Light: nuScenes provides information about the location and orientation of traffic lights on the map. However, information about the dynamic state of the traffic lights (*i.e.* the light colour) is not available.
- Pedestrian Crossing: Area where pedestrians are legally permitted to cross the road.
- Walkway: Area next to a road where pedestrians are protected from other vehicles on the road, also known as pavement.
- Carpark Area: Any area where vehicles can park, whether in a designated parking lot or by the side of the road.
- Road Segment: A section of road within a drivable area, which includes details about the presence of intersections.
- Road Block: Group of adjacent lanes that go in the same direction.
- Lane: Part of the road where vehicles drive in a single direction. Lanes have a *type* which denotes whether vehicles or bicycles are allowed to navigate through them.
- Intersection: Area where multiple lanes intersect.
- Road Divider: A divider that separates one road block from another.
- Lane Divider: A divider that separates lanes pointing in the same traffic direction.

Each element is represented on the map as either a polygon or a line. An example of the map view of a scene is illustrated in Fig. 4.3.

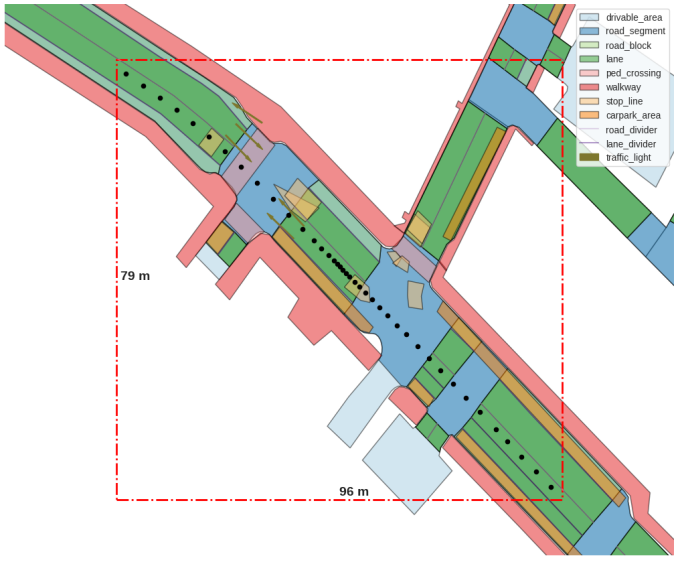


Figure 4.3: Map view of a scene, with the ego vehicle's path indicated by the black dots.

Chapter 5

Methodology

Given the limitations of the state-of-the-art methods and the need for explanations in AVs (Section 2.4), we propose a novel approach for explaining driving behaviour: Policy Graphs with Theory of Mind.

5.1 Policy Graph Formalisation

Policy Graphs (PGs) are a method for representing the behaviour of opaque agents, which in the context of this thesis, consist of autonomous vehicles. A PG is defined as a directed graph $G = (V, E)$ representing the behaviour of an agent in an environment, where V is the set of vertices representing the agent's state S in the environment, and E is the set of directed edges between vertices, representing transitions between states, and therefore, the agent's actions A. This definition focuses on the agent's policy ($P(a|s)$) and how it affects the environment ($P(s'|s, a)$).

PGs are built by observing the agent's behaviour and translating it into a set of state-action trajectories within an environment. State and action spaces are often discretised to make the process feasible in complex environments, simplifying the representation while preserving key features of the agent's behaviour.

PGs were at first introduced by Hayes *et al.* [8], to produce explanations in natural language predicates about robot behaviour, and have proven to provide explanations for simple single and multi-agent scenarios [43]. According to the taxonomy for XAI methods introduced in §2.3, PGs have the following properties:

- *Post-hoc*: An additional model is created to explain the original opaque model.
- *Model-agnostic*: PGs can be used to provide explanations for all types of agents, as they do not require access to the agent's internal model, but they rely only on observations of actions and states.
- *Global and Local*: PGs provide both overall explanations of the behaviour of the opaque agent and explanations of its local decisions.

Concerning the time horizon of the explanation, this initial formulation of PGs has no information about the goals, desires and values of the agent, focusing instead on immediate, reactive explanations. Since understanding the agent's behaviour requires not only knowing which action it takes in a given state but also the purpose behind that action, this limitation can be addressed by introducing notions of beliefs, desires and intentions into the graph, transforming PGs from a *reactive* to a *proactive* model [11]. More details about beliefs, desires and intentions will be explained in §5.3.

5.2 Policy Graph Design

The process of deriving explanations starts by collecting observations of the agent, whether from real-world interactions or simulations within the environment. Following this, one or more discretisers are proposed to represent the agent's states. To build a policy graph, it is crucial to use an abstract representation of the state that captures features of both the agent and the environment relevant to understanding the agent's behaviour. Each state is discretised using predicates in natural language, ideally adhering to the following properties [9]:

- Metric State Space: The state space should allow the computation of similarity between states by defining a distance function.
- State Space Scalability: The size of the resulting state space should be manageable while still allowing the agent to map new observations to previously encountered states.
- Desire Representation: The resulting state representation should enable the formal representation of desires.
- Interpretability: The resulting state representation should be interpretable by the end user.

After defining a proper set of discretisers, the associated PGs are built. There are two possible approaches for building the graph:

- Greedy: The agent takes the most probable action for each state. Consequently, not all agent interactions are present in the graph, but only the most probable action from each node. The output is a directed graph.
- Stochastic: All the agent interactions are added to the graph for each state. Therefore, each node has multiple possible actions, each with an associated probability. The output is a multi-directed graph.

In §2.1, we introduced interpretability and reliability as principles for a good explanation. Reliability refers to how accurate and truthful the explanation is; to be fully reliable, an explanation must precisely represent the model's operations, capturing the low-level details of its algorithms. Interpretability, on the other hand, refers to how easily a human, especially non-technical, can understand the explanation, and it often requires simplifying the model's complex mechanisms into more human-readable terms. This trade-off becomes apparent when comparing discretisers and selecting the optimal one. For this reason, the resulting PGs are evaluated through static metrics (Sections 5.4.1 and 5.4.2), which help to understand the complexity of the PG representation and to have an idea of the interpretability and reliability of the model. It is possible to loop through these steps until a representation with an acceptable balance between interpretability and reliability is achieved. Finally, the explainee registers hypothesised desires into the PG, from which intention metrics that validate these hypotheses can be obtained and give direct values of interpretability and reliability (Section 5.4.3). It is possible to iteratively propose new desires or refine existing ones, as well as adjust the discretisation, until the resulting explanations are satisfactory.

5.3 Theory Of Mind in PGs

To make a PG *proactive*, thus capable of extracting the underlying motivations and objectives of the agent, we incorporate concepts from the Theory of Mind. Theory of Mind (ToM) is the ability to conceptualise, represent, and reason about mental states and behaviours [44]. It is referred to as a "theory" because the behaviour of individuals, such as their actions or statements, is the only aspect that can be directly observed.

Direct access to another individual's mental state is not possible; thus, the characteristics of their mental states must be inferred. It is generally assumed that others possess minds similar to one's own; this assumption allows individuals to attribute beliefs, desires and intentions to others, thereby facilitating the prediction and explanation of their actions [45]. *Beliefs* represent the information state of an agent, its perceptions of the world. The term "belief" is used over "knowledge" given that what the agent believes may not necessarily align with reality and may change over time. *Desires* denote the motivational state of the agent, representing the objectives that the agent aims to achieve. *Intentions* are the plans or specific actions that the agent commits to in order to fulfil its desires. For example, consider an individual that wants to go out for dinner. While at home, the person believes that the restaurant is open (belief) and wants to indulge in something delicious (desire). Based on this belief and desire, the individual decides to go out specifically to go to the restaurant for dinner (intention).

Drawing inspiration from human cognitive processes, this understanding can be a foundational element for designing and implementing models that offer meaningful explanations of opaque agents. These agents can be conceptualised regarding beliefs, desires, and intentions; and interpreting intent is key to understanding their decision-making process [10].

5.3.1 Desires

There are several possible ways to formalise a *desire* [9]: reaching or staying in states where some qualities hold (*e.g.* keeping front vehicles at a distance greater than zero), executing an action in such states (*e.g.* stopping when approaching a red traffic light), or performing a particular transition between world states (*e.g.* overtaking and returning to the initial lane). In this study, we focus on the second type of desires. Action desires can be defined as a tuple $< S_d, a_d >$, where S_d represents a discrete state region ($S_d = \{s \in S | s \vdash d\}$), indicating that state s satisfies the condition for the desire d. The action a_d denotes the desirable action within that state region. This definition can be further expanded to account for desires that can be satisfied by executing one among a set A_d of possible actions drawn from the desire state region, represented as $< S_d, A_d >$. Moreover, desires can also be defined using conditional formulas, such as $< \{s \in S | \varphi(s)\}, A_d >$ where $\varphi(s)$ represents a condition that must be satisfied by one or more predicates of the state s.

After formally defining desires, it is possible to compute relevant information about them within a policy graph: the *desire probability* and the *expected action probability*.

The *desire probability* is the probability of finding oneself in a state where one can fulfil the desire d by performing the action a_d. It can be computed as:

$$P(s \in S_d) = \sum_{s \in S_d} P(s) \tag{5.1}$$

This probability represents how likely it is for the agent to be in a state where it can fulfil a desire d by performing action a. If this value is zero, it means that the opportunity to fulfil the desire does not exist; if the value is high, it means there are many states where the agent has the opportunity to fulfil the desire. This probability is typically low, indicating that the agent is often not in a state where fulfilling the desire is feasible. This can be due to several reasons, such as instances where the agent's conditions rarely align with those necessary for the desire to be achievable (*e.g.* in the case of a diversified dataset) or a wrong desire formulation.

The *expected action probability* is the probability of performing a desirable action when being in the state region S_d. In the case of a single desirable action a_d, it can be computed as:

$$P(a_d | s \in S_d) = \sum_{s \in S_d} P(a_d | s) * P(s) / P(s \in S_d) \tag{5.2}$$

In case of a set of desirable actions A_d, the probability can be computed as:

$$P(a \in A_d | s \in S_d) = \sum_{a \in A_d} \sum_{s \in S_d} P(a | s) * P(s) / P(s \in S_d) \tag{5.3}$$

A high value indicates that when conditions are favourable (when the agent is in a desirable state), the desire is actually fulfilled. On the other hand, a low value can be due to several reasons, including the need to review the discretisation (by adding or modifying predicates), inadequate desire formalisation, or the presence of external factors that prevent the agent from fulfilling the desire, even when it is theoretically in a state to succeed.

As mentioned earlier, the agent is often not in a state where it can directly fulfil a desire, which makes its behaviour difficult to interpret. To address this issue, intentions are introduced in the next section.

5.3.2 Intentions

Intentions can be formalised as the sum of probabilities of all possible paths starting from one state and reaching any state where the agent can fulfil the desire, and it is fulfilled [9]. Formally, let $\mathcal{P}(s, d)$ be the set of paths starting from state s and arriving at any state $s' \in S_d$. The intention to commit to such desire d can be computed as:

$$I_d(s) = \sum_{p \in \mathcal{P}(s,d)} P(a_d | last\_state(p)) * P(p) \tag{5.4}$$

where $P(p)$ is the probability of traversing path p, as computed by the PG:

$$P(p) = \prod_{s', a, s \in p} P(s', a | s) \tag{5.5}$$

From the set of paths $\mathcal{P}(s, d)$, we exclude those paths that fulfil the desired midway. Namely, a path p is excluded if it reaches the state s' by passing through another state $s'' \in S_d$. The

reasoning behind this exclusion is that once the desire has been fulfilled in s'', further fulfilments along that path should not be counted. For example, consider a self-driving car agent that wants to reach two destinations, A and B. Suppose the agent has a 100% probability of successfully reaching destination A, and immediately afterward, it has a 100% probability of successfully reaching destination B. While the car fulfils both desires (driving to destination A and then to destination B), it would be incorrect to claim that the agent has a 200% probability of fulfilling its intention to "drive to a destination". Instead, it must be 100% before arriving at destination A, and 100% after arriving at destination A.

Furthermore, to handle the potentially infinite number of paths, the computation is done backwards, starting from S_d (where the desire can be fulfilled) and recursively propagating intention updates to the parent states.

The intention value $I_d(s)$ represents the probability that a given desire d will be fulfilled in a particular state s. However, as the intention value decreases, the uncertainty around whether the desire will actually be fulfilled increases. To mitigate this, a commitment threshold $C \in (0, 1)$ is introduced, which sets a minimum probability to assess how much the explainee trusts the agent's intent to fulfil a desire. For any state s where $I_d(s) \leq C$, the agent is not considered to be committed to fulfilling the desire. On the other hand, if $I_d(s) > C$, the agent is interpreted as having some intention to fulfil d.

Higher values of C increase the reliability of the explanation, as only states where the agent has a high probability of fulfilling the desire will be considered as states where the intention is active. This ensures that intentions attributed to the agent are more likely to be followed through. Lower values of C improve interpretability by attributing intentions to a broader range of states, making more of the agent's behaviour interpretable. However, this comes at the cost of *reliability*, as some of these intentions may not be actually fulfilled.

To quantify the balance between reliability and interpretability, the previous desire metrics (desire probability and expected action probability) are complemented by two intention metrics dependent on the parameter C. These metrics, discussed in §5.4.3, are the *Intention probability* and the *Expected Intention probability*.

The computed intentions can be used to answer explainability questions, such as *"What do you intend to do in this state?"* and *"How do you plan to fulfil it?"*. The first question is addressed by looking at the intention value $I_d(s)$: given an input state s, we return the desires with attributed intention $I_d(s) > C$. For instance, if the vehicle is moving slowly, aligned in its lane, and oriented to the right with no nearby obstacles, the answer might be *"I intend to perform a Lane Change to the Right"*. To answer the second question, about how to fulfil this objective, the answer can be obtained by retrieving the most optimal path from the input state to the state where the desire is fulfilled. For example, to perform a Lane Change to the Right from the input state, the car might *gas and turn right*, transitioning to a state where it is moving at a medium speed, is on a road divider and oriented to the right. There are no vulnerable road users, stop areas, zebra crossings, traffic lights, or objects nearby, and the vehicle is at the end of a block. From this new state, the vehicle can successfully perform a *Lane Change to the Right* by *turning right.*

5.4 Evaluation

To evaluate the effectiveness of the proposed framework, we need to verify how the integration of PGs and ToM contributes to teleological explanations of AV behaviour. The assessment focuses on three aspects: first, whether the PG accurately represents the behaviour of the driver and the associated state transitions; second, how well it generalises to plausible yet previously unseen driving scenarios; and third, whether the defined desires and intentions of the agent are executed as intended, as well as the extent to which they account for the observed behaviour. To address these aspects, we consider, respectively, three main evaluation criteria: entropy, trajectory likelihood and intention metrics.

5.4.1 Entropy

Entropy evaluates how informative the PG is, in other words, how much the current state determines the following action and state. It is computed as follows:

$$H(s) = H(s', a|s) = -\sum_{s', a \in \{s', a : P(s', a|s) \neq 0\}} P(s', a|s) * \log_2 P(s', a|s) \tag{5.6}$$

Entropy measures whether the design of the PG incorporates relevant information for predictive purposes. It indicates the degree of uncertainty about the next state-action pair given the current state: the higher the entropy, the greater uncertainty about the agent's behaviour and the environment. In our work, entropy is used to compare different discretisations and select the optimal discretiser by looking at the one that minimises entropy, thus achieving lower uncertainty in action and future state predictions.

The equation can be decomposed into $H(s) = H_a(s) + H_w(s)$, reflecting two factors: action entropy $H_a(s)$ and future state (or world) entropy $H_w(s)$, defined as follows:

$$H_a(s) = H(a|s) = -\sum_{a \in \{a:P(a|s)\neq 0\}} P(a|s) * \log_2 P(a|s) \tag{5.7}$$

$$H_w(s) = H(s'|s,a) = -\sum_{a \in \{a:P(a|s)\neq 0\}} P(a|s) * \sum_{s' \in \{s':P(s'|s,a)\neq 0\}} P(s'|s,a) * \log_2 P(s'|s,a) \tag{5.8}$$

Overly simple graphs with a low number of different discretised states show higher action uncertainty $H_a(s)$, making the outputs less reliable (though more interpretable due to their simplicity). On the other hand, a larger PG may result in lower $H_a(s)$ because the discretised states are more specific, making the actions more predictable given a state. However, larger PGs also increase the possibilities for $P(s'|s,a)$, thereby raising $H_w(s)$.

These entropy metrics are used to represent the entropy of a single node and can be extended to the full graph by taking the expected value $E(H_x(s)) = \sum_s P(s) * H_x(s)$, for $H(s), H_a(s), H_w(s)$.

Besides entropy, various structural features contribute to assessing the graph structure, such as graph density and number of weakly connected components[1]. For a directed graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, edge density can be measured as $D = \frac{|E|}{|V|(|V|-1)}$ where $|E|$ is the number of edges and $|V|$ the number of vertices. When analysing the PG's structure, avoiding excessively dense graphs is ideal. Driving scene trajectories might generate highly specific states based on numerous variables such as exact positions, velocities, orientations, *etc.*, and if the state space is overly detailed, even slight differences can result in different states, complicating the agent's ability to map new observations to previously encountered states.

In the context of a PG representing driving scenes, the existence of a single-node connected component is quite rare unless it represents a trivial or degenerate case, such as a state with no meaningful transitions or one that is entirely isolated due to an error or misrepresentation in the model. Typically, states in a PG are interconnected through transitions, either because you can reach them from other states or because you can transition to other states. Therefore, each state should be part of a large connected component, indicating its role in the driving process. If a state does not belong to such a component, it may instead be part of a completely recurrent component. For instance, a car that remains on a motorway indefinitely or a vehicle that is stopped at a traffic light throughout the scene without any transitions. Since these scenarios lack trajectory transitions and are not worth analysing, weakly connected components with only one node are excluded from the PG.

Nonetheless, this entropy formulation has some limitations, as it assumes that the uncertainty associated with selecting a specific action concerning others has a comparable impact on the agent's behaviour. However, there may be critical states (as discussed in [46]) where selecting a particular action has a significantly greater impact on the agent's behaviour.

5.4.2 Likelihood of Trajectory

The second main evaluation criterion is the *likelihood of trajectory*. Trajectory likelihood assesses how well the PG generalises to plausible but unseen driving scenarios. This measure is useful to compare PGs with different discretisations and helps to identify the one that maximises the likelihood of the observed trajectories.

We start with a dataset $\mathcal{D}$ consisting of n episodes (scenes) of state-action trajectories from our agent. This dataset is divided into two subsets: $\mathcal{D}_{train} = \{d_1, d_2, \ldots, d_m\}$, which contains the state-action trajectories from m scenes for training; and $\mathcal{D}_{test} = \{d_{m+1}, d_{m+2}, \ldots, d_n\}$, which contains the trajectories for $n-m$ scenes used for testing. The PG is built on the training episodes.

The likelihood function $\mathcal{L}(PG, d_i)$ is a measure of how well the PG represents the observations from a test scene i. We decompose our analysis into two parts: actions and future states. As regards actions, the likelihood for i_{th} test scene is defined as:

[1] A weakly connected component in a directed graph is a maximal subgraph in which there is an undirected path between any two vertices.

$$-\ell_a(PG, d_i) = -\sum_{(s,a)\in d_i : P_{PG}(a|s)\neq 0} \log P_{PG}(A = a \mid S = s)$$

This measure can be understood as the probability of the actions that the agent took in the test data, compared to the actions that the agent would have taken following the policy graph. The lower, the closer the policy actions match the actual actions of the opaque agent. If s or $a|s$ is not in the model (*i.e.* the PG has never seen this state or action in the training scenes), a possible strategy consists of assigning a small probability ϵ (*e.g.* 0.00001) to these events to avoid negative infinity in the negative-log-likelihood calculation.

The future state likelihood is defined as follows:

$$-\ell_w(PG, d_i) = -\sum_{(s,a,s')\in d_i : P_{PG}(s'|a,s)\neq 0} \log P_{PG}(S' = s' \mid A = a, S = s)$$

This metric is used to assess the probability of reaching a state that the agent actually took in the data, compared to the state that the policy graph would have taken us. The lower, the closer the transition modelled by the PG is aligned with the one from the opaque agent. Also, if the PG has never seen this state transition, we assign a small probability to ensure the $-l$ remains finite and manageable. The obtained results for each test scene are combined by computing the average, assuming that all scenes have the same length. For both $-\ell_a$ and $-\ell_w$, the average across scenes in $\mathcal{D}_{test}$ is:

$$\overline{-\ell_x} = \frac{1}{|\mathcal{D}_{test}|}\sum_{d_i\in\mathcal{D}_{test}} -\ell_x(PG, d_i)$$

In this study, this criterion is used to evaluate the generalisation capabilities of PGs in reflecting the vehicle's behaviour under adverse visibility conditions (§7.2.3).

5.4.3 Intention Metrics

Given the limitations of entropy about the criticality of actions in certain states, Gimenez-Abalos *et al.* [9] propose metrics that take into account the agent's desires and intentions. These metrics evaluate whether the hypothesised and formulated desires are accurate and whether any unexplained behaviour arises that requires further analysis.

The first intention metric is the *Intention Probability*, the probability of being in a state where the intention to fulfil a desire is greater than a certain threshold C, $C \in (0, 1)$. Let $S(I_d)$ be the set of states where the agent is attributed as having the intention I_d greater than C. The intention probability can be defined as $P(s \in S(I_d))$. Additionally, we consider the set of states where the agent is attributed with having *any* of the considered desires as its intention. Given the set of desires D, this can be formalised as $S(I) = \{s \in S | \exists d \in D : I_d(s) > C\}$.

The second metric is the *Expected Intention Probability*, the probability that once attributed, an intention is going to be fulfilled. It is computed as:

$$\mathop{\mathbb{E}}_{s\in S(I_d)}(I_d(s)) = \sum_{s\in S(I_d)} I_d(s) * \frac{P(s)}{P(s \in S(I_d))} \tag{5.9}$$

In case of *any* intention, the probability changes to:

$$\mathop{\mathbb{E}}_{s\in S(I)}(I(s)) = \sum_{s\in S(I)} \max_{d\in D} I_d(s) * \frac{P(s)}{P(s \in S(I))} \tag{5.10}$$

A lower commitment threshold results in a higher intention probability, as it allows more states to be attributed with intentions. This improves interpretability by making more of the agent's behaviour interpretable. However, some intentions may remain unfulfilled, reducing the reliability of the explanations. On the other hand, a higher threshold increases the expected intention probability, meaning states attributed with intentions are more likely to fulfil the associated desires. This improves the reliability of the model. If no commitment threshold provides a satisfying balance between interpretability and reliability, it suggests that the formulated desires do not accurately capture the agent's behaviour.

Intention metrics are used to select the optimal discretiser. To address the challenge of selecting the optimal discretiser and commitment threshold, a ROC-like curve is used, which plots the progression of intention probability (interpretability) and expected intention probability (reliability) in having any of the considered desires as intention, as the commitment threshold changes. The optimal discretiser is then chosen based on the highest area under the curve (AUC), and the optimal value of the parameter C is chosen as the value with the highest values of interpretability and reliability.

It is important to consider that there might be cases where the results of the optimal discretiser differ between entropy (Eq. 5.6) and AUC. In such cases, the choice depends on the priority of the explainee: if certainty in predicting actions or world states is more important, the discretisation minimising entropy H is preferable. On the other hand, if accurately capturing intentions is the priority, the discretisation with the highest AUC should be chosen.

Chapter 6

Experiments

The construction of a PG starts by observing the agent's behaviour and mapping it to a set of trajectories of discrete states and actions (Section 5.1). The first step involves converting all scenes from the processed dataset into trajectories, where each scene follows the structure $(s_t, a_t, s_{t+1}, a_{t+1}, \ldots, s_{t+n}, a_{t+n})$. To achieve this, we propose several discretisers for the state space, following the principles detailed in §5.2, along with an action extraction method to discretise transitions between states. After converting each scene into a state-action trajectory, the graph is built by mapping states to nodes and actions to edges. The PG construction is performed using the Python library *pgeon* [43]. In this work, a stochastic policy graph is built as opposed to a deterministic one. This allows the agent to consider multiple potential actions and states, rather than following a fixed path. After building the graph, the variability of the expected behaviour of the ego vehicle in different scenes is assessed using the metrics introduced in §5.4. If the results are not satisfying, this process can be repeated iteratively, allowing for refinement by exploring alternative discretisations.

6.1 Graph State Variables

In this section, we describe the set of semantically rich features ("predicates") chosen to provide an abstract representation of the state of the ego vehicle. These predicates capture relevant features of both the agent and the environment and are defined using natural language terms to ensure they are interpretable by the explainee. They are defined using the state variables $(x, y, v, a, \psi, \delta)$ and information from the nuScenes map API.

6.1.1 Velocity and Rotation

Two relevant and straightforward predicates that compose the discretised state are *Velocity* and *Rotation*. These predicates are derived directly from the continuous state variables velocity v and steering angle δ, which are discretised following the logic described in Table 6.5. The thresholds for discretisation are identified by consulting the auxiliary DriveLM-nuScenes dataset [47]. The dataset consists of annotated question-answer pairs about nuScenes, organised in a graph that links images with driving behaviour. The DriveLM-nuScenes dataset is selected for its semi-rule-based labelling approach. Here, most data is manually annotated, and the annotators conduct multiple rounds of rigorous quality checks. The labelling process is explained in detail in [47]. Since both velocity and orientation of the ego vehicle are labelled using this approach, we rely on these labels to discretise the values of v and δ. Fig. 6.1 shows the distributions of steering angles by different steering behaviours as annotated in DriveLM-nuScenes. Fig. 6.2 shows the speed distributions by different speed labels from the same dataset. The thresholds are selected based on first and third quartiles, with exceptions made for states when the vehicle is not moving (*Velocity* = *Stopped*) or not turning (*Rotation* = *Forward*). In these cases, we opted for a larger chunk to ensure accurate categorisation, given the sensor variability in measurements.

6.1.2 Ego Vehicle Position and Progress

Differently from the continuous state variables v and δ, the vehicle's global coordinates (x, y) are not directly mapped to a discretised predicate. The reasons are the following:

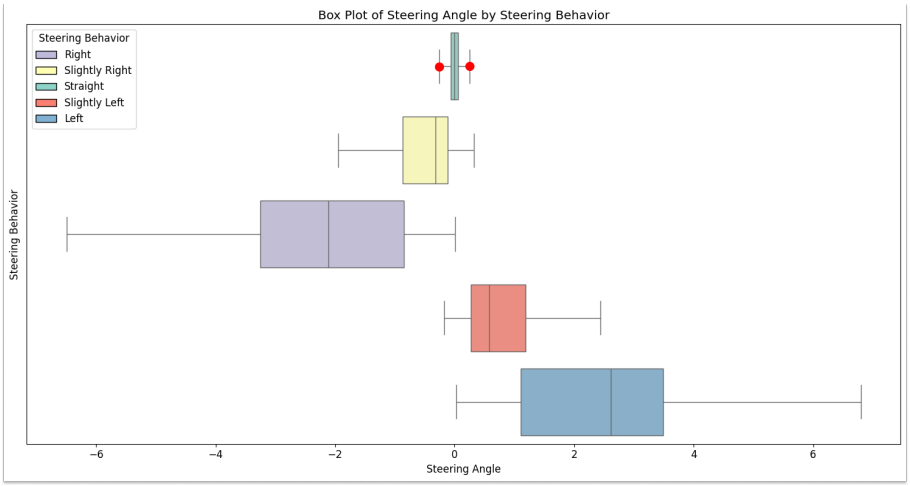


Figure 6.1: The steering angles of the ego vehicle in DriveLM-nuScenes. Each box plot represents the distribution of steering angles for a particular steering behaviour of the ego vehicle. Red dots have been added to the plot to indicate specific thresholds for discretising the steering angles.

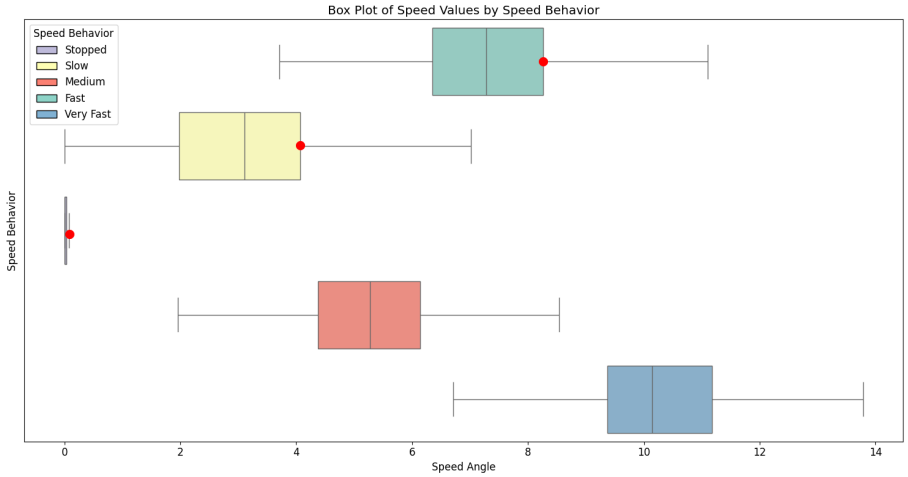


Figure 6.2: Speed values of the ego vehicle in DriveLM-nuScenes. Each box plot represents the distribution of velocity values for a particular speed behaviour of the ego vehicle. Red dots have been added to the plot to indicate specific thresholds for discretising the velocity.

- High dimensionality: Since the locations of the driving scenes are sparse (US and Singapore), the state discretisation may result in a very high-dimensional space.
- Generalisation: Considering position coordinates may limit the ability of the policy graph to generalise to unseen states since the learned policy may only perform well in states in similar positions to those encountered when building the graph.
- Robustness: Directly using position coordinates may make the policy graph sensitive to slight changes in position among states, leading to instability in the learned policy (*e.g.* two states are similar, but the position is slightly different. Consequently, the similarity is not captured by the policy).

Nevertheless, the vehicle's spatial context is useful if combined with environmental data. For instance, the vehicle's position relative to an intersection or a road divider can provide valuable information. In this regard, we extract three predicates that will be part of the discretised state: *BlockProgress*, *LanePosition* and *NextIntersection*. First of all, the drivable area is identified on the map based on the given position (x, y). If the agent falls outside of the drivable area, such as a car parking or a paviment, no meaningful block progress or lane position can be determined. Therefore, the value assigned to these predicates would be *None*.

6.1.2.1 BlockProgress

The predicate *BlockProgress* describes the advancement of a vehicle within the lane it is currently on. This progress is quantified by dividing the lane into three equal chunks, each representing a distinct phase of traversal: the *start*, *middle* and *end*. For situations where the agent lies within an intersection, the value *Intersection* is assigned to *BlockProgress*. An example of this logic is represented in Fig. 6.3.

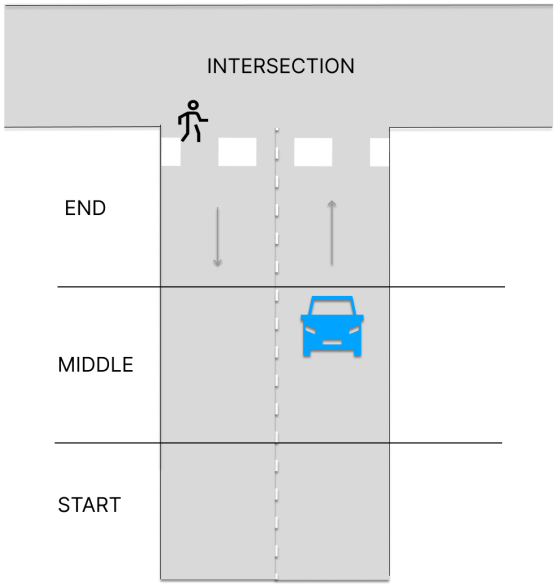


Figure 6.3: Example of different values for the predicate *BlockProgress*. The blue car represents the ego vehicle, and the value of *BlockProgress* in this case would be *Middle*.

6.1.2.2 LanePosition

The predicate *LanePosition* describes the lateral placement of the vehicle within the road infrastructure. It provides information regarding the vehicle's alignment with the current direction of travel, indicating whether the vehicle is oriented in the same direction, opposite to it, or positioned at the centre of the road (*e.g.* on a lane line or road divider). An example is illustrated in Fig. 6.4.

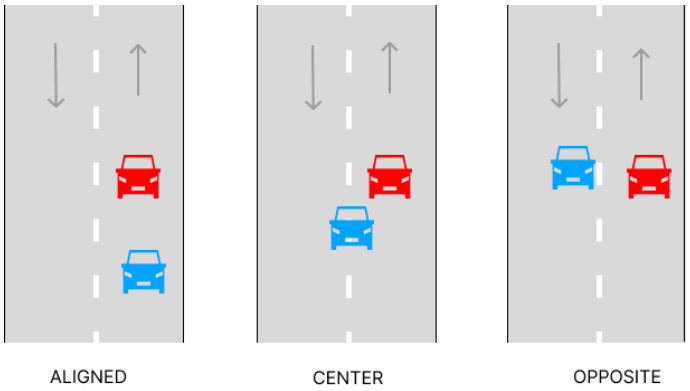


Figure 6.4: Example of different values for the predicate *LanePosition*. The figure shows three stages of the ego vehicle (blue car) overtaking another vehicle (red car), and the different lane positions for each stage.

This predicate is important to identify and understand manoeuvres such as lane changes and overtaking. The assignment of the predicate's value begins by checking if the ego vehicle intersects lane or road dividers. To accomplish this, the ego vehicle is represented as a rectangle centred at position (x, y) and oriented according to its yaw angle ψ, with dimensions approximating the height and width of the true vehicle. The process involves iterating through all road and lane

dividers in the area where the ego vehicle is currently driving to check for intersections with this oriented rectangular representation.

If any intersection between the vehicle and a divider is detected, the value *Center* is assigned to the predicate. If no intersections are found, the value of *LanePosition* is assigned as either *Aligned* or *Opposite*, depending on the direction of travel of the lane in which the vehicle is situated. This assignment is carried out through the following steps:

1. Identify the current or closest lane (within 2m radius) based on the vehicle's position (x, y)
2. Retrieve the arcline path of this lane. The arcline path is a mathematical description of a lane centerline using straight line segments and curved arcs to represent the lane.
3. Compute the closest point on this path to the vehicle's current position and determine a direction vector t considering consecutive points on the arcline path. This direction vector, representing the direction of the lane at the closest point, is then normalised to obtain a unit vector $u = \frac{v}{||v||}$.
4. Compute the vehicle's heading direction as a unit vector h based on its yaw angle ψ in radians.
$$h = \begin{pmatrix} \cos(\psi) \\ \sin(\psi) \end{pmatrix} \tag{6.1}$$
5. Take the scalar product $u \cdot h$ of the lane's direction vector and the vehicle's heading vector, quantifying the alignment between the vehicle's heading and the lane's direction. The results range from -1 to 1, with values closer to 1 indicating alignment with the lane's direction, values closer to -1 indicating the opposite direction, and values near zero suggesting the vehicle is nearly perpendicular to the lane.
6. Depending on the result of the scalar product and a specified threshold, the vehicle's direction of travel is classified as *Aligned*, *Opposite* or *None* relative to the lane. The value *None* is assigned if the current or closest lane to the vehicle has a direction more or less perpendicular to that of the ego car. An empirical threshold $\epsilon = 0.3$ is set for this classification.
$$\begin{cases} Right & if \quad u \cdot h > \epsilon \\ Left & if \quad u \cdot h < -\epsilon \\ None & if \quad u \cdot h \in [-\epsilon, \epsilon] \end{cases}$$

For situations where the agent lies within an intersection, additional processing is required due to the complexity introduced by intersections. Given that the agent is at an intersection rather than a lane, the first step involves identifying the closest lane to the agent and then repeating the approach from step 2. A limitation of this method is that, when searching for the closest lane, the algorithm could potentially return a lane that is not relevant or safe for determining the vehicle's intended manoeuvre.

6.1.2.3 NextIntersection

The predicate *NextIntersection* stores information about the agent's intention at the next intersection (if it exists). This predicate has four possible values:

- *Left*: The ego vehicle will turn left at the next intersection.
- *Right*: The ego vehicle will turn right at the next intersection.
- *Straight*: The ego vehicle will continue straight through the next intersection.
- *None*: There is no intersection in the future path of the ego vehicle.

The approach used to determine the vehicle's action at the next intersection involves assessing the action taken when being at the intersection and then propagating this information back to the previous states. The action at an intersection is determined by comparing the vehicle's position and heading before entering the intersection with its position and heading, after exiting it. The following variables are considered:

- (x_0, y_0): The coordinates of the vehicle's position just before it enters the intersection. This is the last point before the vehicle starts navigating the intersection.

- (x_1, y_1): The coordinates of the vehicle's position at the very beginning of the intersection. This is the first point within the intersection area.
- (x_n, y_n): The coordinates of the vehicle's position at the end of the intersection. This is the last point within the intersection area.
- (x_{n+1}, y_{n+1}): The coordinates of the vehicle's position just after it exits the intersection. This is the first point after the vehicle has completely navigated through the intersection.

Let v_{pre} represent the direction and magnitude of the vehicle's movement from just before the intersection to the start of the intersection. Let v_{post} represent the direction and magnitude of the vehicle's movement from the end of the intersection to just after the intersection. They are defined as follows:

$$v_{pre} = \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix}$$
$$v_{post} = \begin{pmatrix} x_{n+1} - x_n \\ y_{n+1} - y_n \end{pmatrix}$$

To determine how the vehicle's direction changes through the intersection, the angle θ between v_{pre} and v_{post} is computed:

$$\cos(\theta) = \frac{v_{pre} \cdot v_{post}}{|v_{pre}||v_{post}|}$$

The computation of θ (or $\cos(\theta)$) is useful to determine the magnitude of the change in direction. This helps to classify the movement at the next intersection as "going straight" or "turning". If $|\theta| < 20°$, then the value assigned to *NextIntersection* is *Straight*. Otherwise, the sign of cross product is used to determine the direction of turn (left or right):

$$\begin{cases} Left & if v_{pre} \times v_{post} > 0 \\ Right & if v_{pre} \times v_{post} < 0 \end{cases}$$

An example is shown in Fig. 6.5.

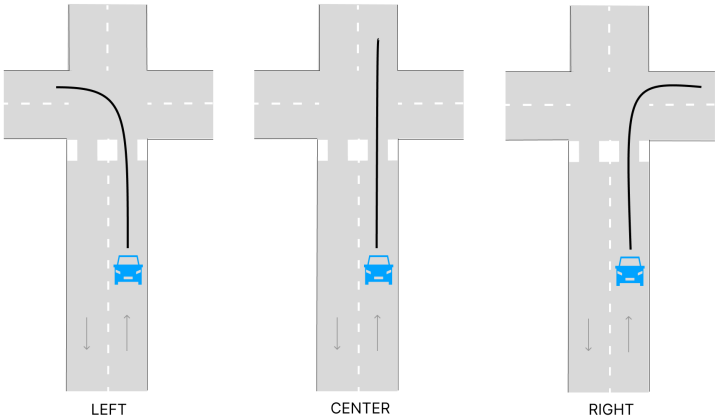


Figure 6.5: Values of the predicate *NextIntersection* representing three potential behaviours of the ego vehicle at an intersection.

6.1.3 FrontObjects

The *FrontObjects* predicate is defined as the presence of objects in the surroundings of the ego vehicle. This predicate provides information about the number of objects explicitly detected by

the front camera of the ego vehicle. Data from the front camera is used due to its wide field of view, which not only covers the area directly in front of the vehicle but also extends slightly to the left and right, offering *good* coverage of the situation ahead. An example of an annotated frame is shown in Fig. 6.6.

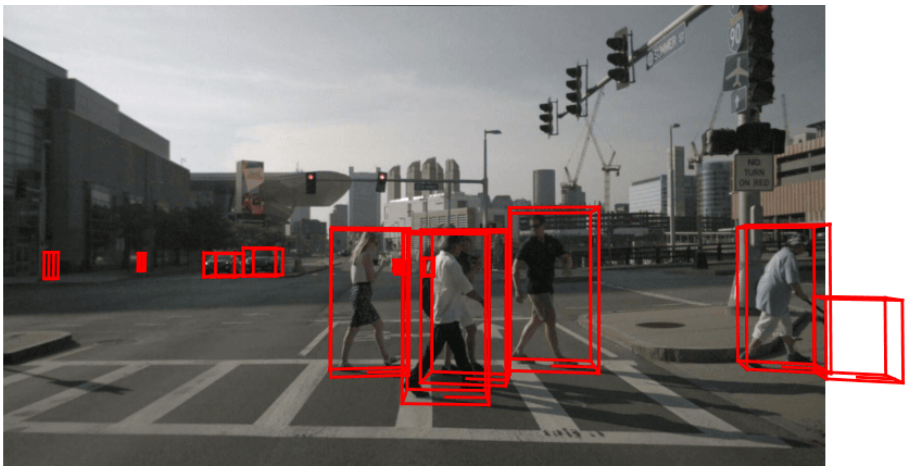


Figure 6.6: Detected objects from the front camera in a frame of scene 553. Each object is contained in a bounding box.

.

The discretisation approach divides the data into two categories:

- No nearby objects: When no objects are detected by the front camera, the predicate is assigned the value *No*.
- One or more objects nearby: If one or more objects are detected by the front camera, the predicate is assigned the value *Yes*.

The distribution of object counts for each category is illustrated in Fig. 6.7.

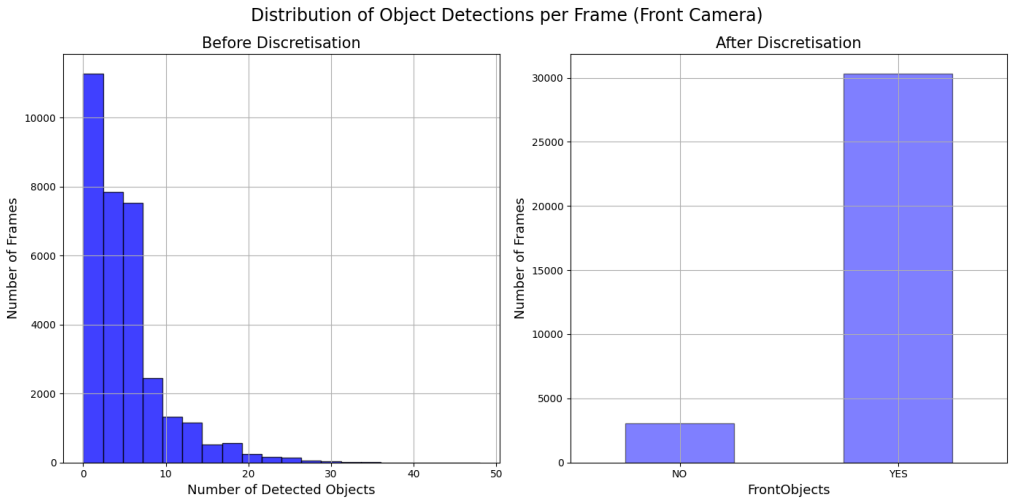


Figure 6.7: Distribution of object detections per frame recorded by the front camera of the ego vehicle, both before and after the discretisation process. The distribution is heavily skewed towards the left, indicating that most frames show fewer detected objects. Few instances contain more than ten detected objects, highlighting the scarcity of scenes in populated areas (*e.g.* schools with children outside, parking slots).

Incorporating data from the front camera is based on the hypothesis that this information can clarify the reasons behind certain behaviours of the ego vehicle that are not otherwise explained by other predicates. For example, while standard stops at traffic lights or stop signs can be identified by the other predicates (Paragraph 6.1.4), scenarios such as avoiding a cone in the vehicle's path or the desire to overtake another vehicle require additional information from the front camera to

understand the vehicle's decision-making process. The initial approach involved integrating data from all six cameras of the vehicle. However, incorporating information about nearby objects from all available cameras significantly increased data complexity as we should add more predicates. This complicates the data processing frameworks upon which our policy graph is constructed and, consequently, the generation of natural language explanations. The more predicates are introduced, the more specific and detailed the policy graph becomes, which in turn demands a larger dataset to ensure adaptability across a variety of scenarios. This is challenging, given the limited size of the dataset. When building our graph we must consider minimal qualitative descriptors so that our graph captures relevant information without growing excessively.

6.1.4 Nearby Road Elements

Three predicates, *IsTrafficLightNearby*, *IsZebraNearby* and *StopAreaNearby*, are included as part of the discretised state to capture the presence of nearby road elements. To assess proximity, a rectangle is centred on the vehicle's current position (x,y) and oriented according to the vehicle's yaw angle ψ, aligning it with the direction of travel. The centre of the rectangle is shifted forward to cover the area directly ahead of the ego vehicle, representing the region drivers typically scan for road signs and elements. The dimensions of the rectangle ($10m$ in width and $12m$ in length), are chosen to ensure it captures road elements not only from the area directly in front of the vehicle but also from neighbouring lanes or the roadside. This provides a sufficient range to detect relevant road elements, even in complex road configurations. Fig. 6.8 illustrates an example of the rectangular area.

6.1.4.1 IsTrafficLightNearby

The predicate *IsTrafficLightNearby* is extracted to store the presence of a traffic light near the ego vehicle. In the nuScenes map expansion, each traffic light is represented by a line, with the starting point of this line considered as the traffic light's location. For every traffic light located within the rectangular area where the ego vehicle is travelling, an assessment is made to determine whether the traffic light is oriented in the direction from which the vehicle is approaching. The vehicle's heading direction is computed as shown in Eq. 6.1. The direction of the traffic light is represented by the direction vector of the line segment denoting it. We compute the alignment by normalising the direction vector and computing the scalar product with the vehicle's heading vector. An alignment value from -1 to -ϵ indicates the traffic light faces the vehicle, influencing its behaviour, and *IsTrafficLightNearby* is set to *Yes*. Values greater than -ϵ (where ϵ is set to 0.1) indicate that the traffic light is not facing the ego vehicle's direction and is unlikely to influence the driving. An example is illustrated in Fig. 6.8.

In addition to individual traffic lights, stop lines associated with traffic lights are also considered. It can happen that a traffic light is far from the ego vehicle (*e.g.* after an intersection), thus not detected within the rectangular area, but the stop line related to the traffic light is close (*e.g.* before the intersection). If the traffic light associated with the stop line faces the ego vehicle, *IsTrafficLightNearby* is set to *Yes*. In all other cases, the presence of a relevant traffic light is excluded. NuScenes dataset does not include explicit information regarding the status of traffic lights (green, yellow, or red), as the map expansion is static.

6.1.4.2 IsZebraNearby

The predicate *IsZebraNearby* is extracted to determine the presence of pedestrian crossings in the vicinity of the ego vehicle. In nuScenes' map, pedestrian crossings are modelled as polygons. If any of these polygons across the city map intersect with the oriented rectangular scanning area around the ego vehicle, the predicate is set to *Yes*. Similarly, the predicate is also set to *Yes* when the scanning area intersects with "turn stops", where the vehicle must yield to pedestrians at a zebra crossing while making a turn. In all other instances, the presence of nearby zebra crossings is excluded. An example is shown in Fig. 6.9.

6.1.4.3 StopAreaNearby

The predicate *StopAreaNearby* is designed to detect the presence of stop signs, yield signs or yielding areas near the ego vehicle. Each stop sign, yield sign or yielding area (such as areas where vehicles must yield before turning, as illustrated in Fig. 6.10) is represented as a stop line. Road signs in nuScenes are not provided with orientations; therefore, if an intersection is between the stop line's polygon and the oriented scanning area, to ensure relevance, we verify if the stop line's roadblock

Figure 6.8: Map view and front-camera view of a frame from scene 103. The red dot marks the ego vehicle's centre in the map view, while the red rectangle represents its scanning area. The arrows represent traffic lights. Some traffic lights intersect the scanning area but are oriented in the opposite direction, so they do not affect the vehicle. However, *IsTrafficLightNearby* predicate is set to *Yes* since the vehicle's scanning area also intersects a stop line (yellow box) related to the traffic lights after the intersection, which are relevant for the vehicle.

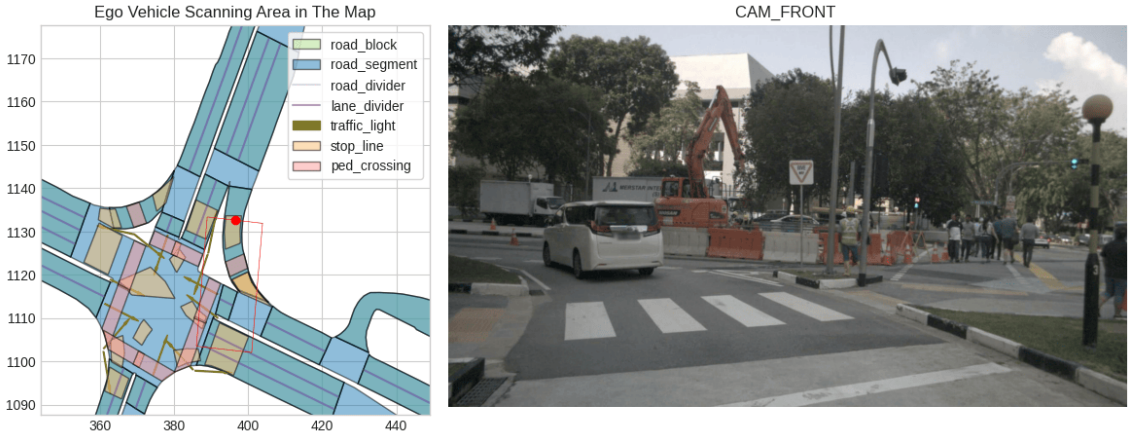


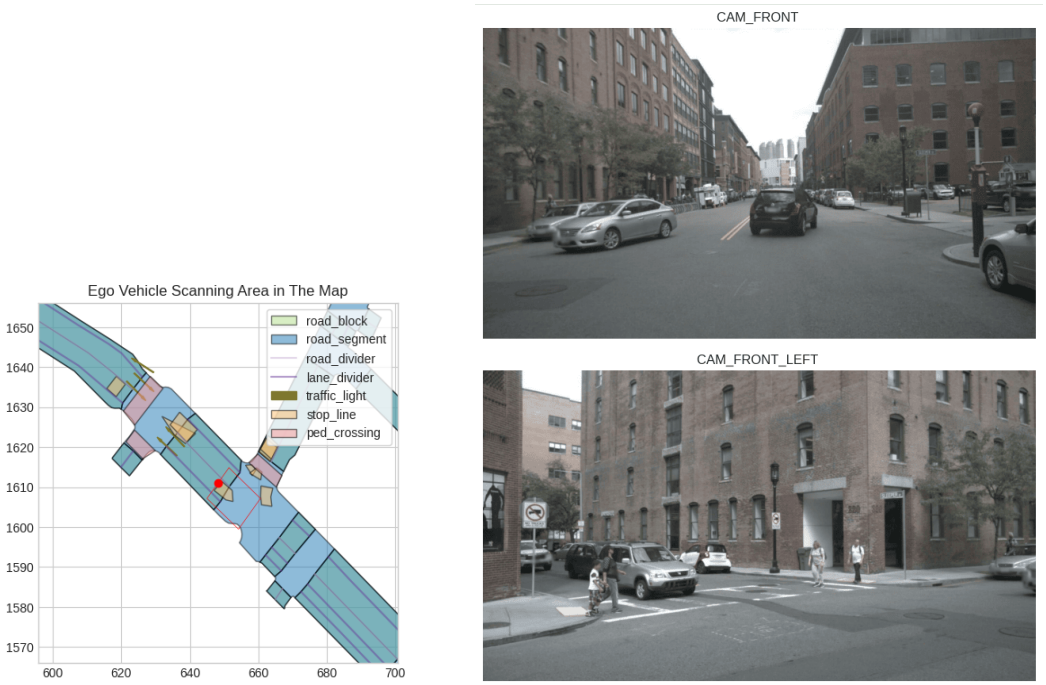
Figure 6.9: Map view and front-camera view of a frame from scene 61. The red dot marks the ego vehicle's centre in the map view, while the red rectangle represents its scanning area. The vehicle is positioned on a stop line for a pedestrian crossing, which sets the *IsZebraNearby* predicate to *Yes*. Additionally, the *StopAreaNearby* predicate is set to *Yield* because a stop line for a yield sign is detected within the scanning area, as shown in the front-camera view.

matches the current roadblock of the vehicle or if the ego vehicle is at a junction (roadblock not defined) . If either condition is met, *StopAreaNearby* is set to *Yield*, *Stop* or *Turn_Stop*, otherwise *StopAreaNearby* is set to *No*. An example is shown in Fig. 6.9.

6.1.4.4 Nearby Vulnerable Elements

Vulnerable subjects include humans, bicycles, motorcycles and scooters with drivers. To identify the presence of these vulnerable road users near the ego vehicle, two predicates are introduced: *PedestrianNearby* and *IsTwoWheelNearby*. These predicates allow for identifying exposed road users who might impact driving decisions and vehicle behaviour. The detection process to identify these subjects relies on front camera data. For predicate *PedestrianNearby*, the analysis focuses on detected humans, defined as any element classified under the 'pedestrian' category in the taxonomy presented in Table 4.1 (*e.g.* adult, child, construction worker, police officer). The discretisation approach divides the data into two categories:

- If no pedestrians are detected nearby, the predicate is assigned the value *No*.
- If one or more pedestrians are detected, the predicate is assigned the value *Yes*.



(a) Map View (b) Camera View

Figure 6.10: Example of turn stop. The car approaches the centre of the road to make a left turn, and yields to other vehicles by waiting on the yellow area marked on the map.

The predicate *IsTwoWheelNearby* refers to bicycles, scooters and motorcycles with riders. Given the limited presence of these subjects in the dataset, even in this case we opted for a simple binary classification: *Yes* (present) or *No* (absent). The distribution of counts for pedestrians and two-wheeled vehicles is shown in Figs. 6.11 and 6.12. For the predicate *PedestrianNearby*, the distribution of values is relatively balanced, indicating that the number of frames where one or more human beings are detected is close to the number of frames where no pedestrians are detected. In contrast, for two-wheeled vehicles, the data shows a significant imbalance. 95% of frames (31,568) contain no two-wheeled vehicles, while only 5% (1,783) of frames include one two-wheeled vehicle. No frames contain more than one two-wheeled vehicle.

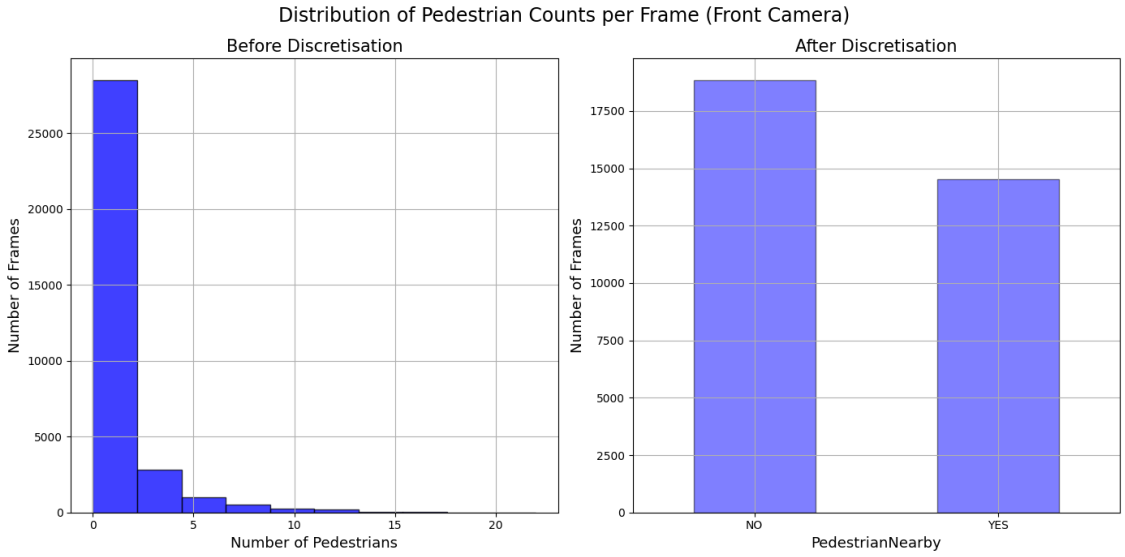


Figure 6.11: Distribution of Counts for Pedestrians from front camera, before and after discretisation. Frames typically detect zero to two pedestrians.

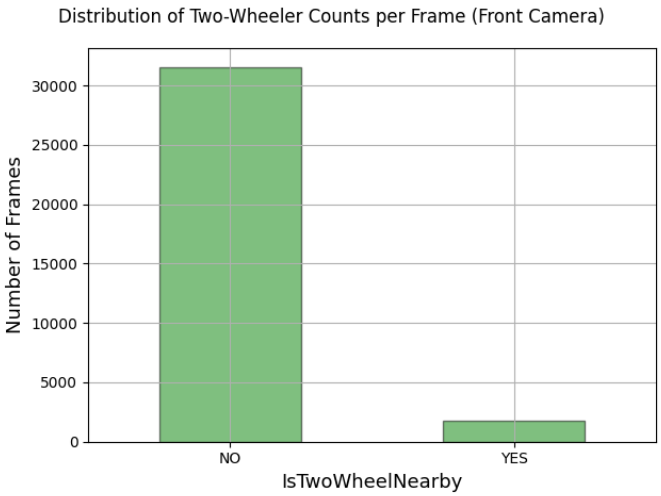


Figure 6.12: Distribution of Counts for two-wheelers from the front camera after the discretisation. The distribution before the discretisation is omitted as they coincide, given that there is only one annotated element in all frames with detected two-wheelers.

6.1.5 Graph State Discretisation

A state discretiser should capture the policy of the agent effectively; avoid being computationally or combinatorially expensive, to prevent an excessive number of possible combinations among all predicate values; and should include predicates that are useful to understand the intentions and desires of the vehicle (Section 5.2). Three main categories of discretisers are defined: D_0, D_1 and D_2. D_0 includes the predicates hypothesised to be the most informative and intuitive for explaining the agent policy. Discretisers D_1 and D_2 expand upon D_0 by incorporating additional predicates, even if their relevance in explaining the agent's behaviour is uncertain. The aim is to explore whether these extra predicates contribute to a better understanding of the agent's behaviour. The process of extracting explanations from the PG is iterative, and improvements or modifications upon the initial hypotheses and implementations may be necessary throughout the process. In the early stages of the workflow (See §6.3.1), discretisers D_0 and D_1 are employed to explain the agent's behaviour. However, the need for further refinement became evident after evaluating the metrics and results obtained from these initial discretisations. This led to the introduction of discretiser category D_2, which is used during the later stages of the workflow (Section 6.3.2). For each discrete category, two variations (a and b) are introduced concerning the possible values of the predicates. Each variation increases in complexity. Table 6.1 and 6.2 give the full description of each discretiser.

Table 6.1: Predicates and possible values for discretisers D_{0a} and D_{0b}

Predicate	D_{0a}	D_{0b}
Velocity	Stopped, Moving	Stopped, Slow, Medium, High
Rotation	Forward, Right, Left	
StopAreaNearby	No, Stop, Yield, Turn Stop	
IsZebraNearby	Yes, No	
IsTrafficLightNearby	Yes, No	
NextIntersection	Right, Left, Straight, None	
FrontObjects	Yes, No	
LanePosition	Aligned, Center, Opposite, None	

Table 6.2: Predicates and possible values for discretisers D_{1a} and D_{1b}. These discretisers are extensions of D_{0a} and D_{0b}, respectively.

Predicate	D_{1a}	D_{2a}	D_{1b}	D_{2b}
D_0 predicates (Table 6.1)	D_0a		D_{0b}	
PedestrianNearby	Yes, No			
IsTwoWheelNearby	Yes, No			
BlockProgress	Start, Middle, End, None			
IdleTime	/	0, 1-4, 5+	/	0, 1-4, 5+

6.2 Action Extraction

To label actions between states, we begin by observing the movement patterns of the ego vehicle through trajectory analysis, with a specific focus on steering manoeuvres, as well as acceleration and deceleration. Actions are then extracted by applying thresholds to key state variables such as velocity, acceleration and steering angle. For instance, a sudden increase in speed may indicate an "acceleration" action. Although these threshold-based methods are approximate, they help to identify various actions based on quantitative changes of the ego vehicle state [48].

In our approach, the state variables used to label the action taken by the agent between states s_t and s_{t+1} are v_{t+1} (velocity), α_{t+1} (acceleration) and δ_{t+1} (steering angle). After extracting the relevant state variable, the vehicle's action is labelled by employing a series of conditional checks to interpret values of velocity, acceleration, and steering angle and map them to specific actions. A summary of the overall approach is presented in Table 6.3, where actions are categorised based on the following criteria:

- *Idling*: If the dynamics of the next state v_{t+1} and α_{t+1} are sufficiently close to zero (within ϵ_v and ϵ_α respectively), the action is labelled as *Idle*.
- *Velocity Modulation*: If v_{t+1} exceeds a positive threshold ϵ_v and α_{t+1} exceeds a positive threshold ϵ_α, the action is labelled as *Gas*. On the contrary, if the velocity exceeds ϵ_v but α_{t+1} is below ϵ_α, indicating deceleration, the action is set to *Brake*. By including both conditions, we avoid misclassifying scenarios where the vehicle speed might exhibit minor alterations that do not constitute meaningful acceleration or deceleration.
- *Direction Modulation*: If δ_{t+1} is less than a negative threshold -ϵ_r, the action is set to *Turn Right*. If δ_{t+1} exceeds the positive threshold ϵ_r, the action is set to *Turn Left*. If δ_t falls within the range ($-\epsilon_r$,ϵ_r), the action is set to *Go Straight*.
- *Combined Action*: If the vehicle performs an action that combines changes in velocity and direction, a combined action of the previous types is assigned (*e.g.* Gas + Turn Right).
- *Default Action*: The default action is set to *Go Straight* if no other conditions are met.

Threshold values are summarised in Table 6.4. The steering and velocity thresholds are derived from Fig. 6.1 and Fig. 6.2, respectively. The extraction methodology is detailed in §6.1.

The acceleration threshold ϵ_α is determined by computing the interquartile range of the distribution of acceleration values, with the first and third quartiles corresponding to -0.3 and 0.3, respectively. As a result, the threshold is set to 0.3.

6.3 Extraction of Explanations

The process of extracting explanations from the PG is iterative, and improvements or modifications upon the initial hypothesis and implementations may be necessary. In the early stages of the workflow (Section 6.3.1), discretisers D_0 and D_1 are employed to explain the agent's behaviour. However, after evaluating the metrics and results obtained from these initial discretisations, the need for further refinement became evident (Section 6.3.2).

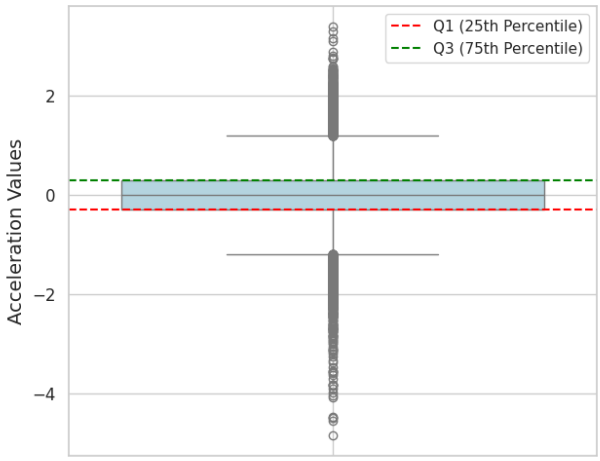


Figure 6.13: Distribution of acceleration values across frames

Table 6.3: Action Determination Logic

Step	**Condition**	**Action**
Idling	$\lvert\alpha\rvert < \epsilon_\alpha \wedge \lvert v\rvert \leq \epsilon_v$	Idle
Velocity Modulation	$\alpha \geq \epsilon_a \wedge v > \epsilon_v$	Gas
	$\alpha \leq -\epsilon_a \wedge v > \epsilon_v$	Brake
Direction Modulation	$\delta \leq -\epsilon_r$	Turn Right
	$\delta \geq \epsilon_r$	Turn Left
	$\lvert\delta\rvert < \epsilon_r$	Go Straight
Combined Action	If Velocity Modulation = *Gas*:	
	$\delta \leq -\epsilon_r$	Gas + Turn Right
	$\delta \geq \epsilon_r$	Gas + Turn Left
	$\lvert\delta\rvert < \epsilon_r$	Gas
	If Velocity Modulation = *Brake*:	
	$\delta \leq -\epsilon_r$	Brake + Turn Right
	$\delta \geq \epsilon_r$	Brake + Turn Left
	$\lvert\delta\rvert < \epsilon_r$	Brake
Default Action	None of the above conditions are met	Go Straight

Table 6.4: Thresholds for Action Determination Logic

Description	**Name**	**Value**
Velocity Threshold	ϵ_v	$0.2 m/s$
Acceleration Threshold	ϵ_a	$0.3 m/s^2$
Steering Angle Threshold	ϵ_r	$0.3 rad$

6.3.1 Early Implementation Stage

After defining the discretisers, the resulting PG is analysed through entropy computation. This evaluation estimates the model's interpretability and reliability, allowing for iterative refinement

Table 6.5: Discretisation logic for predicates *Velocity* and *Rotation*

Predicate	**State Variable**	**Unit**	**Condition**	**Category**
Velocity	Velocity (v)	m/s	$v \leq \epsilon_v$	Stopped
			$\epsilon_v < v \leq 4.1$	Low
			$4.1 < v \leq 8.3$	Medium
			$v > 8.3$	High
Rotation	Steering Angle (δ)	Radians	$\delta \leq -\epsilon_r$	Left
			$-\epsilon_r < \delta \leq \epsilon_r$	Forward
			$\delta > \epsilon_r$	Right

if the trade-off is not acceptable. Following this initial evaluation, the hypothesised desires are introduced into the PG, from which we can obtain intention metrics that validate these hypotheses and give direct indicators of reliability and interpretability. If the explanations derived from this process are found to be unreliable or insufficient, the user can modify the existing desires or formulate new ones. The initial iteration of this process is detailed in this section.

6.3.1.1 Entropy

For each discretiser, entropy $H(s', a|s)$ (Eq. 5.6) is computed to identify the representation that achieves the lower uncertainty in action and future state predictions. Entropy values and graph properties for each discretiser are displayed in Table 6.6.

Table 6.6: Entropy values and graph properties for each initial discretiser (WCC = Weakly Connected Components). The best entropy value is marked in bold casing.

Discretiser	$\|V\|$	$\|E\|$	D	WCC	H	H_a	H_w
D_{0a}	475	3275	0.0145	1	**2.32**	1.55	**0.77**
D_{1a}	1675	8567	0.0031	1	2.63	1.41	1.23
D_{0b}	785	4755	0.0077	1	2.34	1.49	0.85
D_{1b}	2555	11104	0.0017	1	2.48	**1.30**	1.17

The graph's density is increasing with the complexity of the discretiser. As the number of predicates and associated values increases, more complex graphs are generated, but there is insufficient data for the model to generalise effectively. For each discretiser there is only one weakly connected component, showing that all states in the PG are interconnected via transitions. This is an important structural quality, as isolated states would indicate either errors, strange behaviours of the agent or irrelevant states that do not contribute to understanding agent behaviour. The action entropy H_a, thus the uncertainty in predicting the agent's actions, decreases as the complexity of the discretisation increases. This aligns with the theoretical explanation: more specific graphs with more states reduce the uncertainty about the agent's following action, producing a more reliable output. On the other hand, world entropy increases with the increasing discretiser complexity. This indicates that as the discretisation becomes finer, the number of possible future states increases, raising uncertainty about future states ($s'|s, a$). D_{0a} has the lowest overall entropy (closely followed by D_{0b}), achieving the lowest uncertainty in action and future state predictions.

6.3.1.2 Desires Formulation

This section formalises the set of desires that hypothetically guide the agent's behaviour. Looking at the scenes, it emerges that the ego vehicle does not have a precise destination since the scenes are very short (approximately twenty seconds), and the final frame could involve idling at a traffic light or waiting at an intersection. Therefore, the vehicle's desire is not merely to reach a destination but rather to cruise while ensuring compliance (adhering to all traffic laws and regulations), maintaining comfort (minimising sharp manoeuvres), and avoiding collisions.

Table 6.7: First formulation of desires

Desire	S_d	A_d
Stop at Traffic Light	IsTrafficLightNearby = *Yes*	Idle
Stop at Stop or Yield Sign	StopAreaNearby $\in \{Stop, Yield\}$	Idle
Stop at Zebra Crossing	IsZebraNearby = *Yes*	Idle
Obstacle Avoidance	FrontObjects = *Yes*	Turn Right Gas+Turn Right Turn Left Gas+Turn Left
Lane Change	LanePosition = *Center*	Gas+Turn Left Turn Left Gas+Turn Right Turn Right
Turn at Intersection	BlockProgress = *Intersection*	Turn Left Turn Right Gas+Turn Right Gas+Turn Left

For each hypothesised desire, we define the corresponding state region (S_d) and the associated actions (A_d) that are intended to satisfy the desire. The initial formulation of these desires is intentionally simple, as it serves as an initial step for later refinement. Each desire can be individually analysed by computing desire and intention metrics. These metrics are critical for evaluating how well the approach can provide satisfying explanations to the explainee. As simpler discretisers do not fully capture all the desires, we focus on discussing the results for a more complex discretiser, D_{1b}, which offers the clearest insights into how desires are triggered across various agent states. The detailed desire and intention metrics for each discretiser and desire are available in the Appendix A.

At this first stage of workflow, we can think of several possible desires:

- Stop at Traffic Light: This desire specifies the agent's requirement to stop when a traffic light is nearby. It can be formalised as the desire to idle when a traffic light is detected in front. NuScenes dataset does not provide the colour of the traffic light, which would be crucial for refining this desire.
- Stop at Stop Sign: This desire specifies the requirement for the agent to halt when a stop sign is detected in front.
- Stop at Zebra Crossing: The agent is expected to stop when approaching a zebra crossing.
- Obstacle Avoidance: When objects are detected in front of the ego vehicle, the agent is expected to turn away from them.
- Lane Change: This desire covers both moving to the left and right lanes.
- Turn at Intersection: An agent's desire to turn when at an intersection.

The initial set of desires is presented in Table 6.7. Once these desires are registered in the PG, we compute desire and expected action probability (Section 5.3.1) to quantify two aspects: the probability that the agent will find itself in a state where it can act to fulfil a specific desire, and the probability that the agent will successfully perform the desired action once in the corresponding desirable state. The computed desire metrics are shown in Fig. 6.14. As expected, most states do not meet the specific conditions required to *immediately* fulfil a desire, resulting in low values for $P(s \in S_d)$. On the contrary, the desire metrics in obstacle avoidance desires exhibit different

behaviours, suggesting a potential error in formalising this desire. Here, the desire probability is consistently higher than the expected action probability across all discretisers. This indicates that while the conditions necessary for immediately fulfilling the desire are frequently met, the desirable action is rarely taken. A likely explanation is that the definition of the desirable state for obstacle avoidance is overly broad. It includes objects that are not on the path of the vehicle, such as pedestrians on the pavement or parked cars, or objects that are far from the ego vehicle (*e.g.* Fig. 6.6), and which do not require avoidance manoeuvres. As a result, the action to avoid these obstacles is not triggered and the desire is not fulfilled.

For instance, if the ego vehicle is driving on the right lane of an urban area, it would frequently detect parked cars on its front right. The current desire formulation suggests that these situations necessitate a turn to avoid the detected objects, which is incorrect behaviour. In this case, the desire probability should be low, reflecting the infrequent need to avoid obstacles, while the expected action probability should be high when such a situation arises. Furthermore, the low expected action probabilities across most desires indicate a need for refinement in the discretisation and desires formulation. Ideally, the expected action probability should be significantly higher, as it represents the likelihood of fulfilling the desire when the agent is in a desirable state and performing the corresponding action.

The desires must be refined to accurately reflect the specific conditions for each scenario. For instance, the vehicle should only idle at a traffic light if the light colour is red. Additionally, the behaviour associated with a stop sign differs from that at a yield area or yield sign; in the latter case, the driver typically does not want to stop unless required due to another road user's presence. Furthermore, the desire for obstacle avoidance could be misinterpreted in scenarios such as making a simple left or right turn when another vehicle is present in front. When turning at an intersection, it is important to distinguish between left and right turns, as the vehicle may exhibit different intentions in each case.

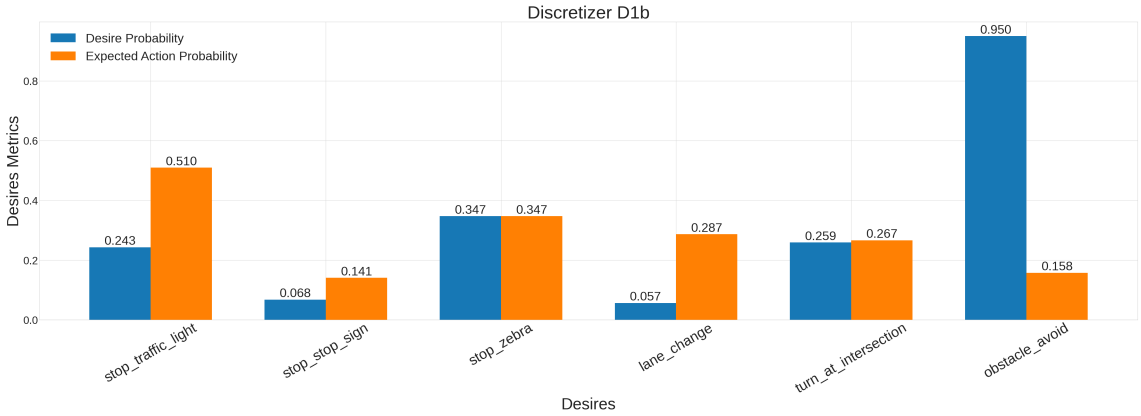


Figure 6.14: Initial desire metrics for discretiser D_{1b}

The results from this initial iteration of the framework underscore the effectiveness of the chosen metrics in identifying flaws in the discretisation and desire formulation. The subsequent improvements are detailed in the final implementation stage described in §6.3.2.

6.3.1.3 Intention Metrics

The intention metrics are illustrated in Fig. 6.15. The probability of manifesting intentions among the specified desires is generally low. At a commitment threshold of $C = 0.5$, the vehicle is attributed with having *any* of the specified desires as its intention only about 32% of the time. This low attribution rate confirms that the current discretisation and definitions of desires do not adequately capture the range of possible intentions, as the desires fail to accurately represent the agent's behaviour. When intentions are attributed, they are fulfilled with 84% certainty. This moderate fulfilment rate indicates that some intentions are correctly identified and acted upon. Still, there may be a misalignment between the fulfilled intentions and the actual behaviour of the vehicle.

Fig. 6.16 illustrates the progression of intention probability (interpretability) and expected intention probability (reliability) as the commitment threshold changes for all four discretisers. The differences across the curves are minimal, indicating limited variation in the predicates or predicate values among discretisers. The best commitment threshold for all discretisers is at

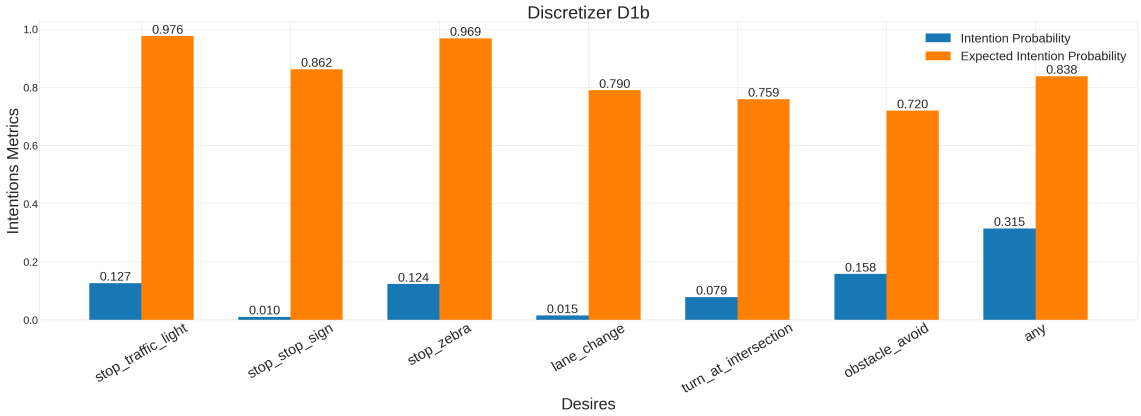


Figure 6.15: Initial intention metrics for discretiser D_{1b}

$C = 0.1$, when almost any non-zero intention probability is considered sufficient to attribute an intention to a state. However, in these instances, the resulting explanations may not be sufficiently reliable. The computed intention metrics confirm the need for redesigning the discretisers and hypothesised desires to better capture the agent's behaviour, as discussed in §6.3.2.

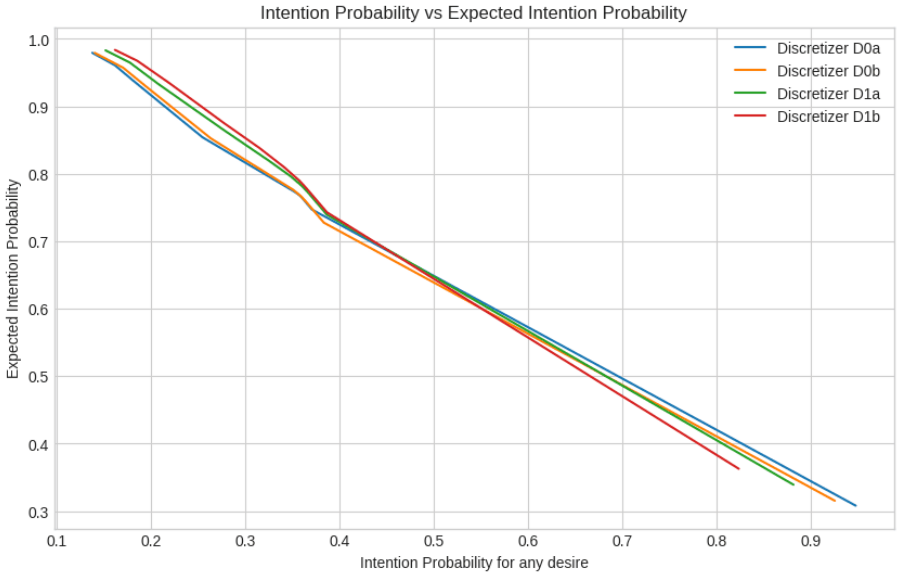


Figure 6.16: Progression of intention probability and expected intention probability as the commitment threshold varies, for all discretisers at the initial stage of the workflow.

6.3.2 Final Implementation Stage

Based on the results that emerged during the first stage of the pipeline, the discretisers are redesigned and the set of desires that hypothetically guide the behaviour of our agent are improved.

6.3.2.1 Improvements over Discretisers

The initial definitions of predicates related to detected elements by the front camera (*FrontObjects*, *PedestrianNearby*, *IsTwoWheelNearby*) require modification due to flaws uncovered in the initial analysis. The high desire probability for the "Obstacle Avoidance" desire (Fig. 6.14) suggests that many annotations can appear in the front camera of a single frame (Fig. 6.6). To extract only relevant front annotations in each frame, for each detected element, the distance from the centre of the ego vehicle to the nearest corner of each bounding box is calculated and used as the object's relative distance. This method is preferred over using the object's centre, as the size of detected objects can vary significantly; for instance, with larger objects like lorries, the centre may not accurately represent the object's proximity to the vehicle. To focus on objects nearby and more likely to affect the vehicle's behaviour, only detections with a distance lower than α from the ego

vehicle's centre are considered, with α set to $12m$. Fig. 6.17 offers a visual example of front camera detections before and after the introduction of α.

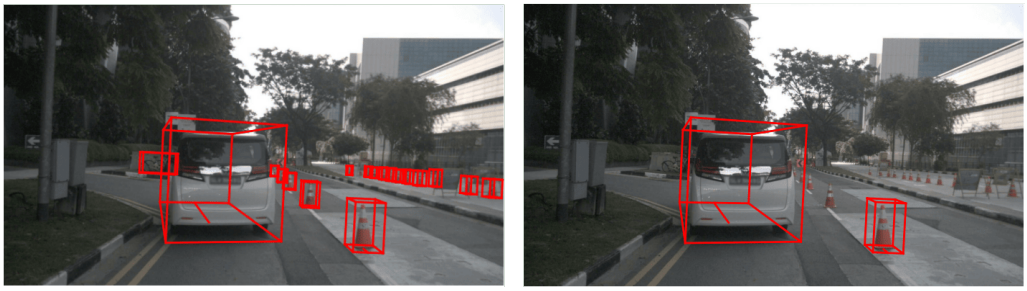


Figure 6.17: Comparison of front camera detections, before and after introducing α

For each predicate (*FrontObjects*, *PedestrianNearby* and *IsTwoWheelNearby*), if no objects of interest are detected by the front camera within α, the predicate value is assigned as *No*. On the other hand, if one or more objects of interest are detected within this range, the predicate value is assigned as *Yes*.

In the case of *FrontObjects*, to further refine the analysis, humans are excluded since they are accounted for by *PedestrianNearby*. Additionally, two-wheeled vehicles without riders are excluded, as they are typically parked on the roadside and do not influence the driving. These exclusions ensure that the focus of *FrontObjects* remains on non-human road elements. The impact of these restrictions is particularly evident in the distribution of values for *FrontObjects* and *PedestrianNearby* (Fig. 6.18). The distribution of values for *IsTwoWheelNearby* remains approximately the same, as the presence of two-wheeled vehicles was already low before introducing α.

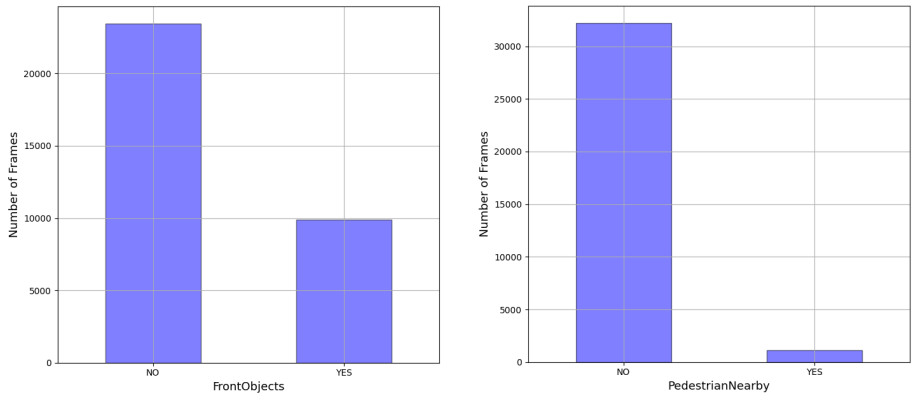


Figure 6.18: Frequency of values for *FrontObjects* (humans and driverless two-wheeled vehicles excluded, $\alpha = 12m$) and *PedestrianNearby* ($\alpha = 12m$). The distributions change drastically compared to the initial version (Figs. 6.7 and 6.11). As expected, the number of relevant front elements shrinks down.

6.3.2.2 Desires

In this second stage of the workflow, a new set of desires has been formalised and grouped into six main categories: cruising, traffic light scenarios, scenarios in the proximity of stop areas, interactions with vulnerable road users, scenarios involving road obstacles (*safe desires*) and unsafe driving behaviours (*unsafe desires*).

Desire and expected action probabilities are computed to validate the hypotheses made during the discretisation process and the formalisation of desires. As simpler discretisers do not fully capture all the desires, we focus on discussing the results for the most complex one, D_{2b}. Each discretiser's and desire metrics are available in the Appendix B.1.

All desires associated to standard driving manoeuvres, such as going straight or turning, are classified under *cruising* and are formalised in Table 6.8. The desire metrics are illustrated in Fig. 6.19. Below is a detailed description of each desire:

- Lane keeping:
  The ego vehicle aims to maintain its moving position within the current lane, oriented forward

and within the lane of the same direction of travel. Furthermore, this desire is active as long as the vehicle is not planning to turn at the next intersection. Desirable actions are actions that do not involve turning, such as accelerating, decelerating, and continuing to go straight at the same pace. The probability of being in a state where lane keeping is feasible is the highest among all desires across different categories. This is supported by a study from Li *et al.* [49], which claims that 73.9% of scenarios in nuScenes involve straightforward driving. The high expected action probability reflects that the desire is almost always fulfilled when the agent is in a favourable state for lane keeping.

- Turn at intersection:
  The desire is activated when the ego vehicle detects that it is within an intersection and is oriented either to the right or left. Desirable actions include executing a right or left turn, eventually combined with either braking or accelerating. The desire is divided into two subcategories: turning right and turning left.

  According to our desire formulation, the probability of being in a desirable state for turning is 4.7% for left turns and 5.5% for right turns. This aligns with dataset driving patterns, where turning at intersections occurs far less frequently than lane keeping, but remains a typical manoeuvre in urban environments. In both cases, the expected desire probability is high, indicating that the corresponding actions are likely to be taken when the vehicle is in the appropriate state for turning.

- Lane change:
  A lane change is a driving manoeuvre that moves a vehicle from one lane to another. It can be divided into:

  - Change onto left lane:
    This desire is activated when the ego vehicle is oriented toward the left and positioned on the lane or road divider, indicating an intention to move into the left lane. Desirable actions include turning left, possibly combined with acceleration or deceleration. Since the dataset is focused on urban scenarios rather than motorways, braking is considered an acceptable action during lane changes, as it can indicate preparation to turn at the intersection. Intersections are excluded from the desirable state region, as lane changes should not occur within junctions.
  - Change onto right lane:
    Similar to the left lane change, this desire is triggered when the vehicle is positioned on the lane or road divider and oriented toward the right. The desirable actions involve right turns, with possible acceleration or deceleration. As with left lane changes, intersections are excluded from the desirable state region.
  - Long lane change:
    This desire represents a specific category of lane changes, where the vehicle executes the manoeuvre gradually and extends its position on the divider for a (relatively) long period. This type of lane change is characterised by a smooth transition over a larger distance, as opposed to sharp or rapid lane shifts (represented by the two previous desires). Differently from the above desires, a critical condition for this desire is the absence of other vehicles in front of the ego vehicle that could interfere with the smooth transition. The desirable state involves all states where the vehicle is oriented forward and on the road or lane divider, with no front objects. If an obstruction is ahead, the vehicle must perform a sharper lane change, falling in the previous two desires. In this case, intersections are also excluded from the desirable state region. The desirable actions involve going straight, using gas or brakes, and maintaining forward movement.

  The probability of being in a desirable state for lane changes is relatively low due to the infrequent nature of such manoeuvres in urban environments. This is reflected in the data, which shows that lane changes occur less often than lane keeping or turning. However, when the vehicle is in a desirable state for lane changing, the high expected action probability indicates that the desired lane change action is frequently performed.

  This desire can be improved with a different formulation since lane changes involve transitioning between states, more than just the execution of an action. Additionally, the current formulation may lead to confusion between the desire to change lanes and the desire to overtake another vehicle. Further improvements could include incorporating state predicates that provide information on the type of divider the vehicle is approaching, such as a white

dashed line, yellow line, or white continuous line. This would allow for the detection of how the driver's intentions shift based on the divider type and help identify actions that do not comply with traffic regulations.

Table 6.8: Formulation of *cruising* desires

Desire	**State Region (S_d)**	**Desirable Actions (A_d)**
Lane Keeping	LanePosition = *Aligned* & Rotation = *Forward* & Velocity ≠*Stopped* & NextIntersection ∉ {*Left*, *Right*}	Gas Brake Go Straight
Turn Left	BlockProgress = *Intersection* & Rotation = *Left*	Turn Left Brake + Turn Left Gas + Turn Left
Turn Right	BlockProgress = *Intersection* & Rotation = *Right*	Turn Right Brake + Turn Right Gas + Turn Right
Change onto Left Lane	Rotation = *Left* & LanePosition = *Center* & BlockProgress ≠ Intersection	Gas Go Straight Brake Turn Left Gas + Turn Left Brake + Turn Left
Change onto Right Lane	Rotation = *Right* & LanePosition = *Centre* & BlockProgress ≠ Intersection	Gas Go Straight Brake Turn Right Gas + Turn Right Brake + Turn Right
Long Lane Change	Rotation = *Forward* & LanePosition = *Centre* & BlockProgress ≠ Intersection	Go Straight Gas Brake

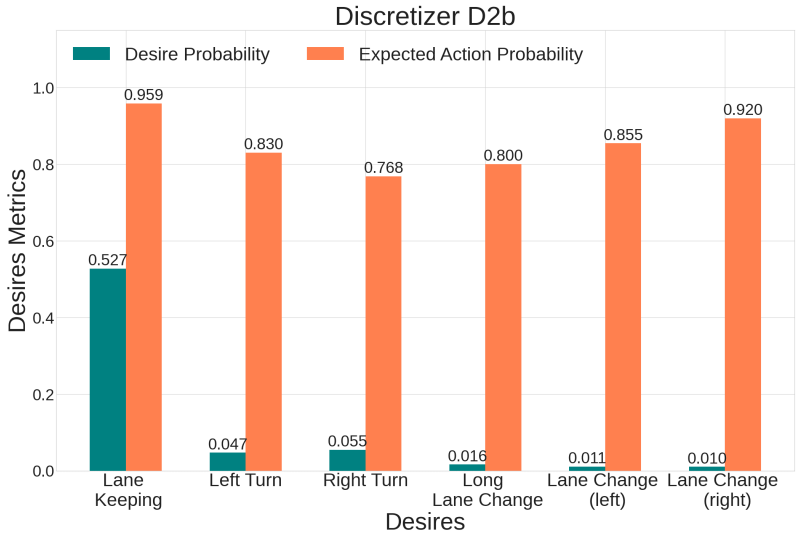


Figure 6.19: Desire metrics for *cruising* desires (D_{2b})

The second category of desires is related to interactions with *traffic light* and is formalised in Table 6.9. Without knowledge of the traffic light's colour status, it is challenging to formalise this behaviour as desirable only when the light is yellow or red. Consequently, these desires remain partially formalised and are used to illustrate how the framework works and how the proposed metrics reveal weaknesses in the discretisation and desire formulations. The computed desire and expected action probabilities for these desires are illustrated in Fig. 6.20. Below is a description of each desire:

- Approach a (red or yellow) traffic light:
  When the ego vehicle is moving at medium or high speed and a traffic light is in close proximity, it is generally desirable to slow down. This is important when the traffic light is yellow or red, as the vehicle must slow down and prepare to stop. However, braking is unnecessary if the light is green, as the vehicle can continue its course without interruption. The results show that the ego vehicle is rarely in a state where it is moving at a considerable speed while approaching a traffic light. In the few instances where this occurs, the probability of the vehicle slowing down because it is approaching a traffic light is low (32.7%). This can be attributed to the lack of information on the traffic light's colour status. If the traffic light colour was known, the behaviour could be more accurately captured in the desire formalisation.

- Waiting at (red) traffic light:
  This desire is triggered when the vehicle is stopped at a traffic light after detecting a red light. The expected behaviour is for the ego vehicle to idle and wait for the light to turn green. The ideal desire formulation would be to idle until the light colour changes, but since the dataset lacks this information, this condition is not applied to the desirable state region. Despite this limitation, the high action probability (97.5%) shows that the ego vehicle often remains idling at traffic lights for an extended time, indicating that in many scenes, the vehicle spends several seconds idling at a red light before it moves again.

Table 6.9: Formulation of *traffic light* desires

Desire Category	**State Region (S_d)**	**Desirable Actions (A_d)**
Approach Traffic Light	IsTrafficLightNearby = *Yes* & Velocity $\in$ {*Medium*, *High*}	Brake Brake + Turn Right Brake + Turn Left
Wait at Traffic Light	IsTrafficLightNearby = *Yes* & Velocity = *Stopped*	Idle

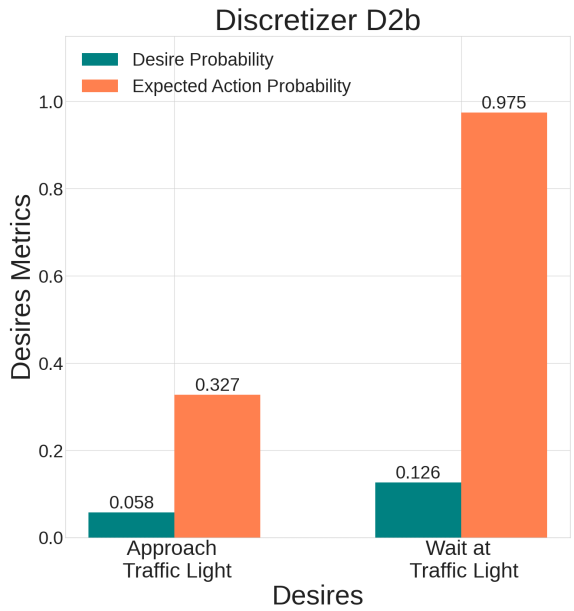


Figure 6.20: Desire metrics for traffic light desires (D_{2b})

Another category of desires is related to behaviours in the proximity of stop areas (turn stops, yield signs and stop signs). The computed desire and expected action probabilities for these desires are illustrated in Fig. 6.22. Table 6.10 summarises the description of each desire:

- Approach a stop sign:
  When approaching a stop sign, the desired behaviour is to come to a complete stop. The state region includes situations where the vehicle is in motion near a stop sign, with no nearby traffic lights that could influence the vehicle's movement. This distinction ensures the stop is governed only by the presence of the stop sign. The desirable action is to slow down or

stop. The expected action probability is not high (45.2 %). This suggests that, in the limited frames where the vehicle is moving in the proximity of a stop sign, it does not stop or brake. This behaviour may be attributed to drivers often performing a "rolling stop", where they eventually reduce speed and check that nobody is in their way without coming to a full halt.

- Wait at stop sign:
  This desire reflects the need to wait at a stop sign to get the right to proceed. The state region includes situations where the vehicle is entirely stopped at a stop sign, with no nearby traffic lights that could influence the vehicle's movement. The ideal action is for the vehicle to remain stationary at the stop sign (until it is safe to proceed). The desire is divided into three different scenarios, all of which are very rare in the dataset (with $P(s \in S_d) \leq 0.001$):

  - Natural waiting time: After the vehicle has just stopped, the driver needs a moment to assess whether the path is clear and it can proceed safely. The key condition on the state region is that the idling time is zero, indicating that the vehicle has just halted and needs a few moments to check its surroundings to determine if it is safe to move. The expected action probability is high (87.5%). However, it would expected to be closer to 100%. This discrepancy may be attributed to drivers frequently performing a "rolling stop".
  - Waiting due to objects in the front: When there are objects in the vehicle's front area, such as other vehicles, obstacles or animals, the desirable action is to remain stopped until the path gets free. The expected action probability for this scenario is 66.7%, indicating that the ego vehicle generally waits when objects block the way, but not always. This may be because the predicate *FrontObjects* includes a broad range of elements, from vehicles to static objects. A possible improvement could be refining this predicate to distinguish between objects, vehicles and animals or even between moving and non-moving objects. Defining more specific predicates rather than just a general one could allow for a more precise analysis.
  - Waiting due to pedestrians nearby: Similarly to the previous case, the vehicle detects pedestrians nearby. Hence, the desirable action is to remain stopped (until it is safe for pedestrians and vehicles to proceed). The expected action probability is 1, meaning that every time the ego vehicle waits at a stop sign with a pedestrian nearby, it consistently idles.

  These reasons for waiting are not mutually exclusive, as the vehicle may need to idle at a stop sign due to a combination of these factors.

- Start from stop sign:
  This desire reflects the driver's aim to proceed from a stop sign once it is safe to do so. It is triggered after the vehicle has been waiting at the stop sign for at least 4 seconds, allowing sufficient time to stop and double-check for traffic. The conditions include the absence of nearby traffic lights, no detected objects in the front and no pedestrians near the vehicle. Additionally, the vehicle must have a planned action to move forward ($NextIntersection \neq None$). This condition is crucial because, in some instances, scenes in nuScenes are truncated at stop signs, traffic lights, or pedestrian crossings, with the vehicle still waiting to pass when the scene ends. By specifying that the vehicle has a planned movement, we ensure the desire corresponds to situations where the vehicle is proceeding.

  Idling time information is introduced since, as described in the previous desires, the driver usually takes a few seconds at the stop line before getting back on the road again. This happens because the vehicle is legally forced to stop, even if the road is free. Initially, no discretiser accounted for the predicate *IdleTime*, which resulted in a desire probability of 0.001 and an expected action probability of 0.256. This low expected action probability indicated the need for revision in both the discretisation and desire formulation, leading to creating a new predicate. This adjustment exemplifies how the iterative improvement process works. By specifying $IdleTime > 0$ as a condition on the state region, we observe an improvement in the expected action probability to 0.355. Despite the improvement, the probability remains relatively low, indicating that other factors influence the vehicle's restarting delay. To investigate further, we focus on the same extended desire state region and compute the expected action probability when the action space is restricted to $A_d = \{Idle\}$. In this case, the expected action probability increased to 0.61, suggesting that 61% of the time, the vehicle continues idling instead of moving forward. This highlights the presence of

additional factors causing hesitation to restart. While the influence of traffic lights, nearby pedestrian crossings, and longer waiting times was tested, no impact was found. A plausible cause for the delay may be undetected front objects that still pose a risk, such as cross-traffic at intersections.

- Yield to vehicles:
  When the ego vehicle approaches a yield sign at the end of a road or a yielding area (turn stop) at an intersection to perform a turn ($NextIntersection \neq Straight$), the ego vehicle's desire is to slow down and eventually stop to yield to the vehicles travelling on the main road and coming from the opposite direction. This is not necessarily a complete stop, but enough to ensure safety and compliance with traffic regulations. The expected action probability is quite high (0.663), though there is space for improvement. Similar to the case of waiting at a stop sign with front objects, the issue could derive from the broad nature of the *FrontObjects* predicate, which includes vehicles, movable/static objects, and animals. In scenes where the vehicle is yielding, it is important to note that the dataset may sometimes cut the scene midway through this phase. For this reason, we allow *NextIntersection* to have value *None*.

  Initially, the desire was tested considering yield signs only (excluding turn stops), and the desire probability resulted in being zero (Fig. 6.21). This outcome highlights either a problem with the discretisation, a wrong desire formulation, or the desire never being brought about and thus satisfied. This case is the latter: by further investigating the dataset, there are only three yield signs in the whole map. By exploring the policy graph, no node has the predicate *StopAreaNearby* set to *Yield*, meaning that the ego vehicle never gets close to one of these few traffic signs. This example shows our methodology's ability to investigate the agent's behaviour.

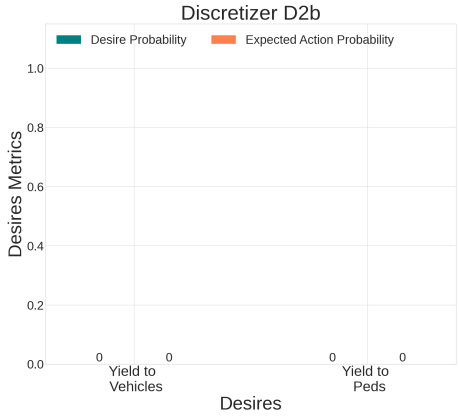


Figure 6.21: Desire metrics for the yielding desires (turn stops excluded).

- Yield to pedestrians:
  This desire mirrors the desire of yielding to other vehicles, but it applies when approaching a crosswalk. If the ego vehicle is turning (either left or right) and encounters pedestrians crossing at the crosswalk, the desirable actions are to slow down or stop to yield. As with the previous case, *NextIntersection* is set to *None*. In instances where the ego vehicle must yield to pedestrians, the probability of slowing down or stopping is relatively high (0.874), indicating that in the few cases where this scenario occurs, the vehicle generally behaves correctly.

Desires related to behaviours in proximity of vulnerable road users are formalised in Table 6.11. The computed desire and expected action probabilities for these desires are illustrated in Fig. 6.23. Below is the description of each desire:

- Let pedestrian cross at crosswalk:
  If a crosswalk is nearby the ego vehicle and pedestrians are detected, the desirable action is to either slow down or come to a full stop. The driver should stop to let the pedestrian cross, but this is not always the case. For example, if the detected pedestrians are not exactly in front of the vehicle (*e.g.* already at the end of the crosswalk or crossing on the other side of the road), it might happen that the vehicle does not stop. To analyse driving behaviour in this scenario, we define two sub-desires:

Table 6.10: Formulation of *stop areas* desires

Desire	**State Region (S_d)**	**Desirable Actions (A_d)**
Approach Stop Sign	StopAreaNearby = *Stop* & IsTrafficLightNearby = *No* & Velocity ≠ *Stopped*	Idle Brake Brake + Turn Right Brake + Turn Left
Wait at Stop Sign	StopAreaNearby = *Stop* & IsTrafficLightNearby = *No* & Velocity = *Stopped* & IdleTime = *0*	Idle
Wait at Stop Sign (obj)	StopAreaNearby = *Stop* & IsTrafficLightNearby = *No* & Velocity = *Stopped* & FrontObjects = *Yes*	Idle
Wait at Stop Sign (peds)	StopAreaNearby = *Stop* & IsTrafficLightNearby = *No* & Velocity = *Stopped* & PedestrianNearby = *Yes*	Idle
Start from Stop Sign	StopAreaNearby = *Stop* & IsTrafficLightNearby = *No* & Velocity = *Stopped* & FrontObjects = *No* & PedestrianNearby = *No* & IdleTime ≠ 0 & NextIntersection ≠ None	Gas Gas + Turn Right Gas + Turn Left
Yield to Vehicles	StopAreaNearby ∈ {*Yield*, *Turn_Stop*} & Rotation ≠ *Forward* & FrontObjects = *Yes* & NextIntersection ≠ *Straight* & BlockProgress ∈ {*End*, *Intersection*}	Idle Brake Brake + Turn Right Brake + Turn Left
Yield to Pedestrians	StopAreaNearby ∈ {*Yield*, *Turn_Stop*} & Rotation ≠ *Forward* & PedestrianNearby = *Yes* & NextIntersection ≠ *Straight* & BlockProgress ∈ {*End*, *Intersection*} & IsZebraNearby = *Yes*	Idle Brake Brake + Turn Right Brake + Turn Left

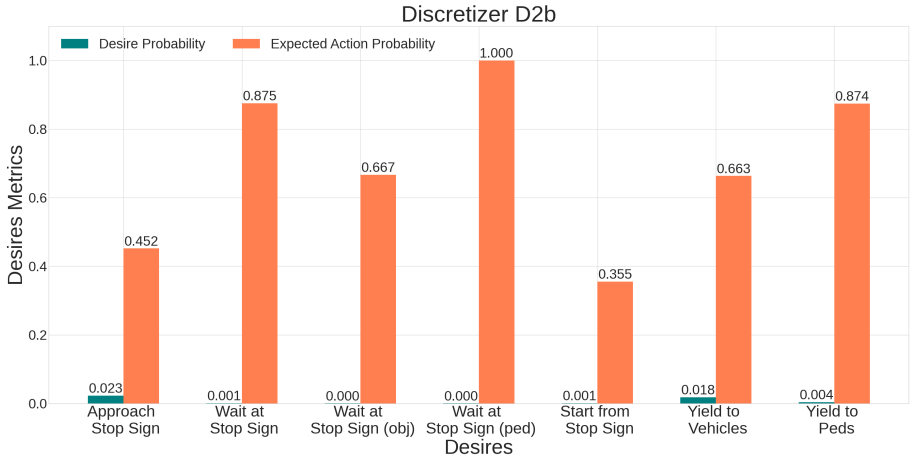


Figure 6.22: Desire metrics for stop area desires (D_{2b})

– Let pedestrian cross at a crosswalk when driving at high speed:
This version of the desire considers only states where the vehicle is going at medium or high speed and detects pedestrians in the proximity. The desirable action is to slow down since drivers at higher speeds need to reduce their speed significantly in the presence of pedestrians to ensure safety, even if they do not want to stop completely.

  - Let pedestrian cross at crosswalk when driving at low speed:
    This version considers the scenario where the vehicle moves slowly, and pedestrians are detected. This formulation is based on the fact that in most jurisdictions, the vehicle must wait until the pedestrian has fully crossed before entering the crosswalk. However, in practice, many drivers only slow down enough to allow pedestrians to pass safely on their side of the road. They may accelerate again as the pedestrian nears the opposite side.

  The metrics show that drivers often do not adequately slow down or come to a full stop when approaching pedestrians at a crosswalk, especially when moving at lower speeds. The expected action probability is low in both high-speed and low-speed scenarios, indicating that the vehicle's behaviour does not always align with the desirable actions. Several factors may contribute to this behaviour: the vehicle might not decelerate or halt if the pedestrian has already crossed their side of the road, if they are beginning to cross from the far side, or in instances where a pedestrian is detected but not actively crossing (for example, standing on the pavement). Additionally, drivers tend to slow down only partially or proceed once the pedestrian is no longer perceived as an immediate concern. This last behaviour is prevalent in real-world driving despite contradicting traffic regulations in many countries.

- Caution with random pedestrians:
  This desire targets situations where pedestrians are jaywalking (crossing roads outside designated areas), when a police officer is directing or standing in the road, or when construction workers are at work. The vehicle's responsibility is to slow down, stop, or take evasive action (turns) to avoid a potential collision. Although such occurrences are rare in the dataset, the expected action probability is high, suggesting that the vehicle does not always respond appropriately when it encounters pedestrians outside of typical crosswalks or stop areas. Possible reasons may be that the ego vehicle detects pedestrians on the pavement or construction workers working on the side of the road that do not affect driving. This result highlights the need for a new predicate to distinguish between pedestrians on and off the drivable area.

- Caution with two-wheel vehicles nearby:
  When the ego vehicle travels at high speed and detects a two-wheeler (such as a bicycle, motorcycle, or scooter) nearby, the vehicle should decrease its speed to be more cautious around these vulnerable subjects. The vehicle is rarely in this desirable state, with a low probability of occurrence ($P(s \in S_d) < 0.001$), as expected by the infrequent presence of two-wheel vehicles in the dataset. The expected action probability of 0.6 indicates that when the ego vehicle is in this state region, it slows down slightly more than half the time.

Table 6.11: Formulation of *vulnerable road users* desires

Desire	**State Region (S_d)**	**Desirable Actions (A_d)**
Pedestrian at Crosswalk in high speed	IsZebraNearby = *Yes* & PedestrianNearby = *Yes* & Velocity $\in$ {*Medium*, *High*}	Brake Brake + Turn Right Brake + Turn Left
Pedestrian at Crosswalk in low speed	IsZebraNearby = *Yes* & PedestrianNearby = *Yes* & Velocity = *Low*	Idle Brake Brake + Turn Right Brake + Turn Left
Jaywalk Response	IsZebraNearby = *No* & PedestrianNearby = *Yes* & Velocity $\neq$ *Stopped*	Idle Turn Left Turn Right Brake Brake + Turn Right Brake + Turn Left
Nearby Two-Wheeler	IsWheelNearby = *Yes* & Velocity = *High*	Brake Brake + Turn Right Brake + Turn Left

The desires related to *obstacle avoidance*, formalised in Table 6.12, address situations where the ego vehicle must avoid objects by moving either left or right. These scenarios' computed desire and expected action probabilities are illustrated in Fig. 6.24. Below is a detailed description of each desire:

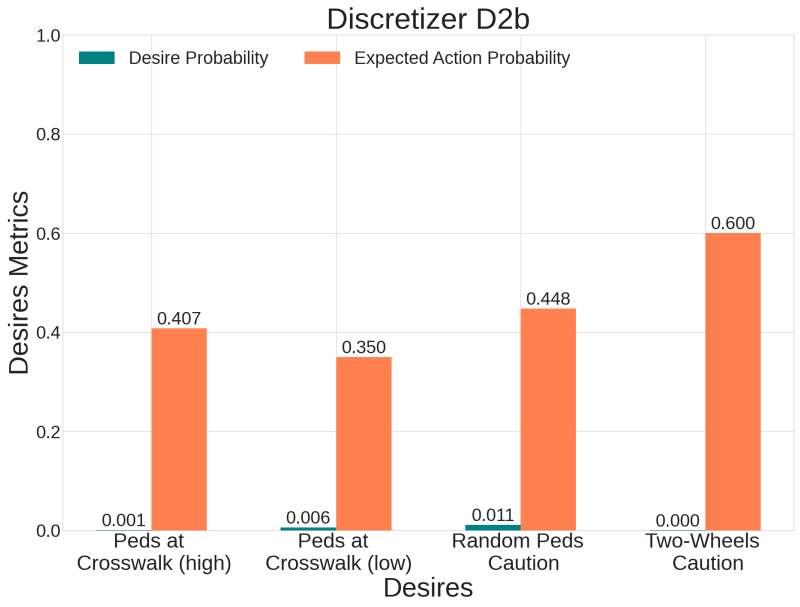


Figure 6.23: Desire metrics for vulnerable road user desires (D_{2b})

- Avoid object by going on the right:
  This desire is triggered when the ego vehicle detects objects (such as cones, debris, or vehicles) in its front path and is rotated toward the right. The desirable action is to execute a right turn to avoid the obstacle. To prevent confusion with a regular right turn at an intersection, we specify that the vehicle should either proceed straight or turn to the left at the next intersection, thus excluding right turns.
- Avoid object by going on the left:
  This desire is similar to the previous one but is triggered when obstacles are detected in front of the vehicle, which is oriented towards the left. In this case, the vehicle ideally steers left to avoid the obstacle.

The expected action probability is high (around 0.81). This indicates that when the vehicle is in the desirable state of avoiding objects, it generally takes the correct action of steering away from the obstacle.

Table 6.12: Formulation of *obstacle avoidance* desires

Desire	**State Region (S_d)**	**Desirable Actions (A_d)**
Avoid Object (Right)	Rotation = *Right* & LanePosition = *Aligned* & FrontObjects = *Yes* & NextIntersection $\neq$ *Right* & Velocity $\neq$ *Stopped*	Turn Right Brake + Turn Right Gas + Turn Right
Avoid Object (Left)	Rotation = *Left* & LanePosition = *Aligned* & FrontObjects = *Yes* & NextIntersection $\neq$ *Left* & Velocity $\neq$ *Stopped*	Turn Left Brake + Turn Left Gas + Turn Left

The final category refers to all undesirable behaviours on the road and should be avoided to ensure safety and compliance with traffic regulations. We rely on PGs to detect whether the driver in the scenes exhibits intentions of such unsafe road behaviours, formalised in Table 6.13. Desire and expected action probabilities are illustrated in Fig. 6.25. The metrics show that these unsafe desires are highly unusual, as the highest desire probability is only 0.041, indicating that the driver is in an unsafe desirable state just 4.1% of the time. Furthermore, the expected action probabilities for this category are generally lower than those for standard desires, confirming that the probability of these unsafe desires being realised is lower than for typical driving desires. Below is the description of each desire:

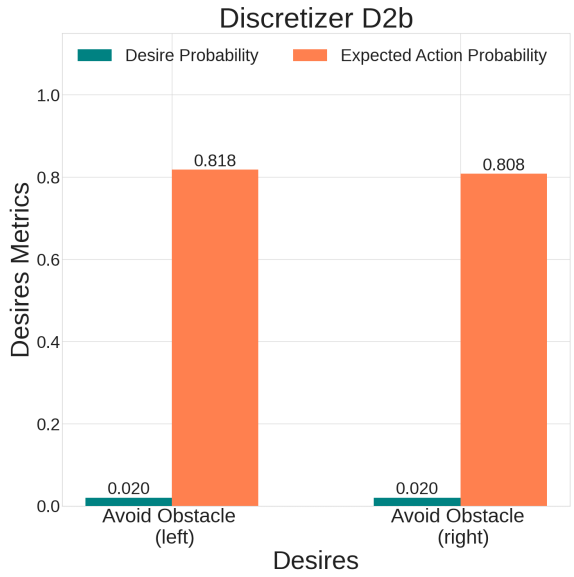


Figure 6.24: Desire metrics for obstacle avoidance desires (D_{2b})

- Ignore random pedestrians:
  The ego vehicle fails to adequately respond to pedestrians who are crossing or is near the ego vehicle but far from designated crosswalks (*e.g.* jaywalking, construction workers, police officers). The formalisation includes that the vehicle is going at medium or high speed and it accelerates or keeps going straight. This behaviour is dangerous in urban environments where random crossing is frequent. The expected action probability is relatively high, indicating a concerning tendency for the vehicle to ignore pedestrians in these situations. A plausible explanation may be the presence of construction workers on the road in several scenes.

- Ignore pedestrians at crosswalk:
  This behaviour is brought about when the ego vehicle approaches a crosswalk with pedestrians nearby, and it keeps going straight or accelerates. We excluded turns from the set of unsafe desirable actions since they could be a way to avoid pedestrians (it is not a desirable behaviour but is better than accelerating or keeping going straight). Depending on the driver's velocity, we differentiate between ignoring pedestrians at high speeds and at low speeds. In both scenarios, the expected action probability is not low, which may be attributable to the bad practice of rolling stops.

- Not yielding to oncoming vehicles:
  This behaviour occurs in situations where the ego vehicle approaches an intersection to turn, and it disregards the requirement to yield to oncoming traffic. The unsafe desirable actions include turning, eventually accompanied by acceleration. The expected action probability is low, indicating that instances, where the ego vehicle does not yield to oncoming objects, are rare. A possible explanation for why this probability is not zero is that the *FrontObjects* predicate may also capture non-vehicle entities such as cones or debris.

- Not yielding to pedestrians:
  This behaviour is characterised by the ego vehicle failing to yield to pedestrians who are crossing the road when performing a turn. The unsafe desirable actions involve all actions in which the vehicle does not reduce speed, instead opting to continue turning through the crossing area and, in some cases, accelerate. The expected action probability is low, indicating that instances, where the ego vehicle does not yield to pedestrians crossing, are rare. A plausible reason why the probability is not zero could be that pedestrians may have already crossed or are starting to cross on the opposite side of the road.

- Overlook presence of two-wheeled vehicles:
  Despite detecting these smaller vehicles, when travelling at high speeds, the ego vehicle does not adequately account for nearby two-wheeled vehicles, such as bicycles, motorcycles or scooters. The low expected action probability of this desire confirms that the driver in the scenes is aware of the vulnerability of two-wheeled vehicles.

- Exit Drivable Area:
  This desire describes the situation in which the ego vehicle leaves the designated drivable area, such as lanes or roads, and enters non-drivable zones (*e.g.* pavements). The dataset contains no instances of the ego vehicle leaving the drivable area, as supported by the desire metrics.
- Ignore Safety Distance:
  This behaviour is observed when the vehicle travels at high speed, detects an element in front, and continues to accelerate, failing to maintain a safe following distance. The expected action probability is low, indicating that it is infrequent that such behaviour is uncommon in the dataset. A possible explanation for why the probability is not zero could be that the dataset also contains a few scenes on motorways, where higher speeds are typical.

Table 6.13: Formulation of *unsafe* desires

Desire	**State Region** (S_d)	**Desirable Actions** (A_d)
Ignore Random Pedestrians	PedestrianNearby = *Yes* & IsZebraNearby = *No* & Velocity ∈ {*Medium*, *High*}	Go Straight Gas Gas + Turn Left Gas + Turn Right
Ignore pedestrians at crosswalk (high velocity)	PedestrianNearby = *Yes* & IsZebraNearby = *Yes* & Velocity ∈ {*Medium*, *High*}	Go Straight Gas Gas + Turn Left Gas + Turn Right
Ignore pedestrians at crosswalk (low velocity)	PedestrianNearby = *Yes* & IsZebraNearby = *Yes* & Velocity = *Low*	Go Straight Gas Gas + Turn Left Gas + Turn Right
Not yield to oncoming vehicles	BlockProgress ∈ {*End*, *Intersection*} & StopAreaNearby ∈ {*Yield*, *Turn Stop*} & Rotation ∈ {*Left*, *Right*} & FrontObjects = *Yes* & NextIntersection ≠ *Straight*	Turn Left Turn Right Gas + Turn Left Gas + Turn Right
Not yield to pedestrians	BlockProgress ∈ {*End*, *Intersection*} & StopAreaNearby ∈ {*Yield*, *Turn Stop*} & Rotation ∈ {*Left*, *Right*} & PedestrianNearby = *Yes* & NextIntersection ≠ *Straight*	Straight Turn Left Turn Right Gas + Turn Left Gas + Turn Right
Overlook presence of two-wheeled vehicles	IsTwoWheelNearby = *Yes* & Velocity ∈ {*Medium*, *High*}	Gas Gas + Turn Left Gas + Turn Right
Exit drivable area	BlockProgress = *None*	All possible actions
Ignore safety distance	FrontObjects = *Yes* & Velocity = *High*	Gas Gas + Turn Left Gas + Turn Right
Not Wait at the Stop Sign	StopAreaNearby = *Stop* & IsTrafficLightNearby = *No* & Velocity = *Stopped* & IdleTime = *0*	Gas Gas + Turn Left Gas + Turn Right
Not Wait at Stop Sign (obj)	StopAreaNearby = *Stop* & IsTrafficLightNearby = *No* & Velocity = *Stopped* & FrontObjects = *Yes*	Gas Gas + Turn Left Gas + Turn Right
Not Wait at Stop Sign (ped)	StopAreaNearby = *Stop* & IsTrafficLightNearby = *No* & Velocity = *Stopped* & PedestrainNearby = *Yes*	Gas Gas + Turn Left Gas + Turn Right

We initially considered also the desire to overtake. By the current formulation of desires $< S_d, A_d >$, the intentions may get confused with the ones of lane change or obstacle avoidance. What distinguishes overtaking is the return to the initial lane after the lane-changing process. This requires a desire formalised as $< S_d, A_d, S_{d+1} >$ which is left for future work. Other driving desires such as u-turns and emergency lane changes (*e.g.* due to ambulance or police presence) are not considered due to limited coverage of scenarios by the dataset.

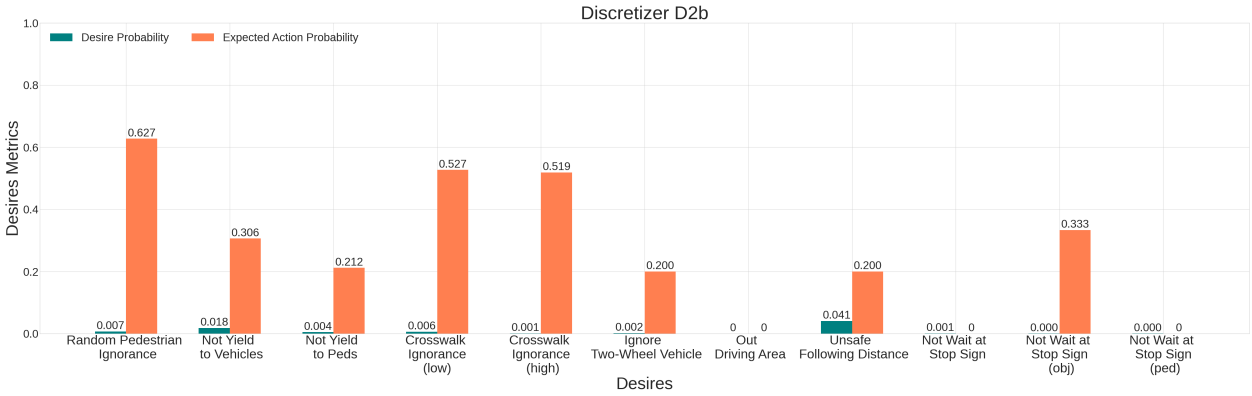


Figure 6.25: Desire metrics for *unsafe* desires (D_{2b}). The metrics show that these unsafe desires are highly unusual, as the highest desire probability is only 0.041, indicating that the driver is in an unsafe desirable state just 4.1% of the time. Furthermore, the expected action probabilities for this category are generally lower than those for standard desires, confirming that the probability of these unsafe desires being realised is lower than for typical driving desires.

6.3.2.3 Intention Metrics

After defining the desires that hypothetically drive the agent behaviour, intention metrics (Section 5.4.3) are computed to evaluate whether the agent shows intentions to satisfy the formulated desires and whether it fulfils them. This analysis helps to validate that the proposed approach offers proactive explanations of the agent's behaviour. The commitment threshold C is set to 0.5. The discussion focuses on the most complex discretiser, D_{2b}.

Fig. 6.26 shows the intention metrics related to cruising desires. Aside for the lane keeping, where the intention probability is moderate, the intention probability for other desires tends to be low. This can be explained by the desire metrics, which show that the vehicle is occasionally in a state where these desires can be satisfied. Nevertheless, the states where the agent is attributed with such intentions are fulfilled with a high degree of certainty, as reflected by the expected intention probabilities close to 1. This pattern is consistent across various desires, and an overview of the intention metrics for each discretiser and desire can be found in Appendix B.2.

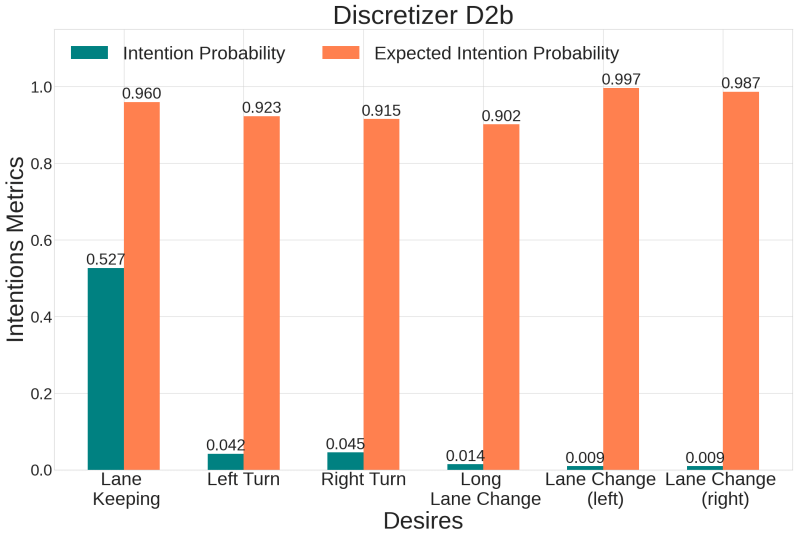


Figure 6.26: Intention metrics for cruising desires (D_{2b})

Table 6.14 provides a summary of the intention metrics for *any* desire, meaning that the agent is attributed with having *any* of the considered desires as its intention. The table also distinguishes between desires categorised as *safe* (*e.g.* yielding to oncoming vehicles) and *unsafe* (*e.g.* ignoring crossing pedestrians). The results indicate that the intention probability is consistently high across all discretisers for both safety and any general desire. This suggests that a large portion of the agent's behaviour is interpretable and that we are frequently able to attribute clear intentions behind the vehicle's decisions. The expected intention probability is also high for all discretisers,

meaning that nearly every attributed intention fulfils the corresponding desire, reflecting the reliability of the generated explanations. In general, interpretability improves with more complex representations. Specifically, using D_{2b}, we can attribute any intentions to the vehicle's behaviour 81.2% of the time, with 95.2% certainty that the attributed intentions are fulfilled.

Instances of unsafe desires are infrequent, as the dataset seldom captures scenes involving unsafe or illegal behaviours. Consequently, the vehicle rarely intends to fulfil these desires, with intention probabilities being very low across all discretisers. At most, only 1.4% of the time we can attribute an unsafe intention to the vehicle, as most intentions relate to safe desires. However, for those rare cases where unsafe intentions are attributed, the probability that the agent fulfils the related desires is satisfying across most discretisers. In particular, for discretisers D_{1b} and D_{2b} attributed intentions of any unsafe desire are fulfilled with an 82.5% certaininty.

Discretizer	**Any Safe Desire**		**Any Unsafe Desire**		**Any Desire**	
	Intention	Exp. Intention	Intention	Exp. Intention	Intention	Exp. Intention
D_{0a}	0.695	0.952	0	0	0.695	0.952
D_{1a}	0.803	0.951	0.007	0.745	0.805	0.950
D_{2a}	0.806	0.951	0.007	0.747	0.808	0.951
D_{0b}	0.699	0.952	0.001	0.648	0.699	0.952
D_{1b}	0.806	0.952	0.014	0.825	0.809	0.952
D_{2b}	0.809	0.952	0.014	0.825	**0.812**	**0.952**

Table 6.14: Intention probability (interpretability) and expected intention probability (reliability) for any safe desire, any unsafe desire, and any desire (safe and unsafe) across various discretisers.

Fig. 6.27 illustrates the progression of intention probability (interpretability) and expected intention probability (reliability) as the commitment threshold changes for any desire and across all six discretisers. All discretisers demonstrate high metrics as the commitment threshold changes, suggesting that the interpretability-reliability trade-off is generally well-balanced. In particular, discretisers D_{1b} and D_{2b} achieve the highest area under the curve, making them the optimal representation for capturing intentions. For both D_{1b} and D_{2b}, the best value for the commitment threshold is found to be 0.1

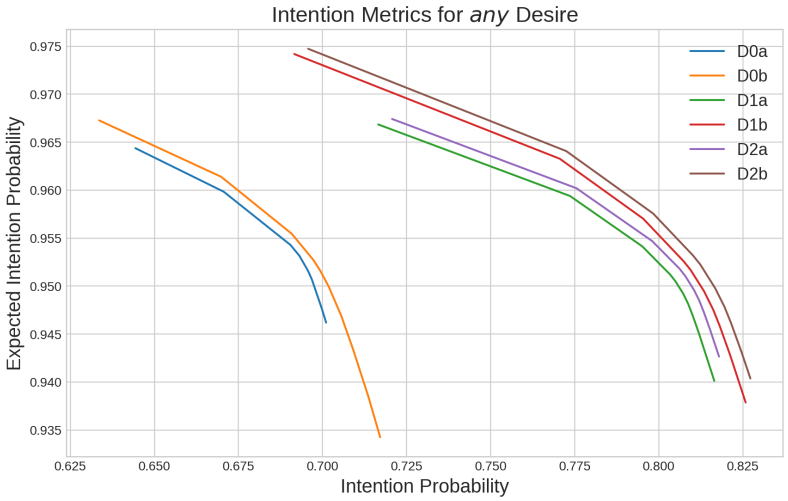


Figure 6.27: Progression of intention probability and expected intention probability as the commitment threshold varies, for all discretisers at the final stage of the workflow.

6.3.2.4 Entropy

Entropy is computed for each discretiser to identify the representation that achieves the lower uncertainty in action and future state predictions. The entropy values and graph properties for each discretiser are presented in Table 6.15.

The action entropy H_a, which reflects the uncertainty in predicting the agent's actions, decreases as the complexity of the discretisation increases, since more specific graphs with more states reduce the uncertainty about the agent's next action. On the other hand, world entropy H_w increases with greater discretiser complexity, indicating that as the discretisation becomes finer, the number of possible future states rises, leading to increased uncertainty about future states.

D_{2a} and D_{2b} have three connected components: one very large and two smaller ones containing only three nodes each. This occurs because there are scenarios where the vehicle remains stopped at a traffic light throughout the scene, resulting in the representation of a single state with no transitions to other states. Since D_{2a} and D_{2b} include a predicate to track idling time with three possible values ($IdleTime \in \{0, 1{-}4, 5\}$), these scenarios are represented as small isolated connected components.

Among the discretisers, D_{0b} has the lowest overall entropy, achieving the lowest uncertainty in action and future state predictions, although most values of the other discretisers are close.

This discretiser differs from the one identified as optimal by intention metrics. The choice depends on the priority of the explainee: if certainty in predicting actions or world states is more critical, D_{0b} is preferable. On the other hand, if accurately capturing intentions is the primary goal, then D_{2b} should be selected. Discretisers D_{1b} and D_{2b} perform well in both evaluations, making them versatile choices.

Table 6.15: Entropy values and graph properties for each final discretiser (WCC = Weakly Connected Components). The best entropy value is marked in bold casing.

Discretiser	$\|V\|$	$\|E\|$	D	WCC	H	H_a	H_w
D_{0a}	562	4096	0.013	1	2.60	1.54	**1.06**
D_{1a}	1389	7722	0.004	1	2.77	1.44	1.33
D_{2a}	1678	8122	0.003	3	2.78	1.43	1.34
D_{0b}	936	5901	0.007	1	**2.58**	1.48	1.11
D_{1b}	2153	10285	0.002	1	2.62	1.34	1.28
D_{2b}	2442	10685	0.002	3	2.63	**1.33**	1.30

Chapter 7

Biases and Behavioural Patterns in NuScenes

The risks posed by biased autonomous driving systems are significant, with a primary source of harmful bias and discriminatory performance often rooted in the data on which these systems are built (Section 3). Real-world driving datasets frequently display significant imbalances, which may lead to the unfair treatment of certain road users. To identify potential biases in such data, we conduct a quantitative analysis of the nuScenes dataset, focusing specifically on class imbalances. In addition to class imbalance, we also examine how driving behaviour is affected by diverse visibility conditions, such as weather and lighting, that are not often well represented in real-world driving scenarios.

7.1 Dataset Analysis

The scenes in the dataset were collected from two countries: the United States, which accounts for 55% of the scenes, and Singapore, which contributes the remaining 45%, thus including both left-hand and right-hand driving scenarios. Furthermore, nuScenes includes a diverse range of situations, from standard environments such as intersections, construction sites, and pedestrian crossings to more rare scenarios involving ambulances and animals. However, 73% of the scenes involve straightforward driving [49], which skews the dataset towards more conventional driving conditions.

Each frame in the dataset is annotated with objects from 23 distinct classes (Table 4.1). Particular emphasis is placed on analysing the frequency of annotations for vulnerable road user classes, including pedestrians, cyclists, and motorcyclists, as their underrepresentation would pose a significant threat to the effectiveness of the vehicle object detection system.

Overall, the *pedestrian* class is well represented, as the dataset includes a significant proportion of pedestrian annotations at 19.05% of the total. Specifically, are 208,240 instances labelled as *adult* pedestrians, which accounts for 17.86% of all annotations, making it the second highest category after the *car* class. However, other pedestrian subcategories, such as children (0.18%), construction workers (0.79%), individuals with personal mobility devices (0.03%), pedestrians with strollers (0.09%), those using wheelchairs (0.04%), and policy officers (0.06%) are severely underrepresented. This disparity highlights the lack of diversity among various human road users, further exacerbated by the absence of key attributes essential for studying bias, such as skin tone.

For the *cyclist* and *motorcyclist* classes, annotations constitute only 2.1% of the total dataset annotations, with a significant portion being instances without drivers (*e.g.* parked bicycles). This low level of representation may negatively affect the model's ability to accurately detect cyclists and motorcyclists.

In addition to *representation* bias, we must also be concerned about *labelling* bias, as class and attribute labelling of detections is a crucial step in influencing biases within the vehicle object detection system. In nuScenes, driving data is sent to an annotation partner that employs expert human annotators and multiple validation steps to achieve accurate annotations [1]. However, the annotation process lacks transparency, making it difficult to assess their methodology. For instance, in nighttime scenes, it is difficult to determine whether the lower number of pedestrians is due to their absence or because they are not accurately detected.

7.2 Impact of Visibility on Driving

Visibility conditions, such as time of day and weather, typically influence driving behaviour. We analyse *how* these factors affect the decisions of the agent using the processed version of nuScenes. The classification of scenes by time of day (day or night) and weather conditions (rain or no rain) is derived from human-annotated descriptions of each scene. Of the analysed scenes, 11.9% occur at night, and 20.6% involve rainy conditions. Scenes are divided into three broader visibility categories: high visibility (daytime with no rain), medium visibility scenes (either night or rain), and low visibility (nighttime with rain). The frequency of scenes in these categories is illustrated in Fig. 7.1, showing that the dataset is skewed towards high visibility conditions.

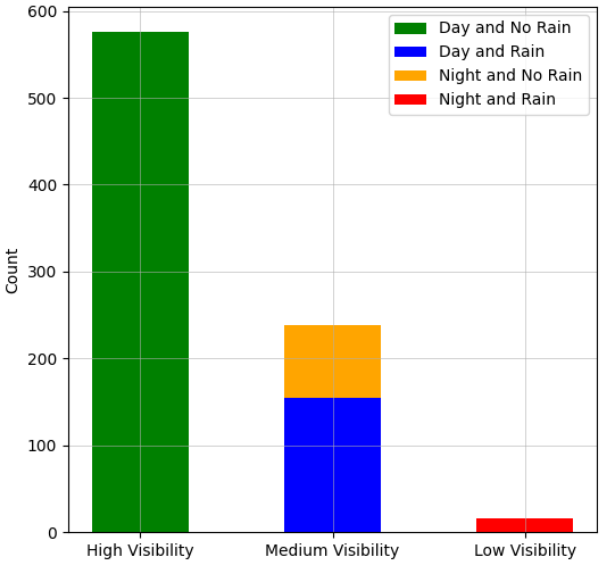


Figure 7.1: Frequency of scenes based on visibility conditions

7.2.1 Weather Conditions

To assess the impact of weather on the agent's behaviour, its intentions are compared between scenarios involving rain and those with good weather. Fig. 7.2 illustrates some of the intention metrics most affected by weather conditions. In bad weather, no instances are observed where the driver intends to approach a crosswalk at high speed when pedestrians are present, highlighting that the driver is typically more cautious under such conditions.

In adverse weather, when the driver has attributed the intention to approach a crosswalk at low speed when pedestrians are present, the expected intention probability is higher than a good weather scenarios. This indicates that intentions to allow pedestrians to cross in bad weather are more frequently fulfilled than in good weather, where rolling stops are less dangerous and tend to be more common.

Furthermore, the vehicle is never in a state where it intends to be cautious about two-wheeled vehicles, probably because these vehicles are less likely to be on the road during rainy conditions. Generally, intentions related to vulnerable road users are less pronounced (very low intention probabilities) in adverse weather than in good weather. This could be attributed to the reduced presence of these vehicles in rainy conditions or perhaps to limitations in the vehicle's visual system to detect them.

7.2.2 Time of Day

To assess the impact of light conditions on the agent's behaviour, its intentions are compared between night scenes and day scenes. Fig. 7.3 illustrates some of the intention metrics most affected by the time of the day. In adverse lighting conditions, when the driver is attributed the intention to approach a crosswalk at low speed in the presence of pedestrians, the probability

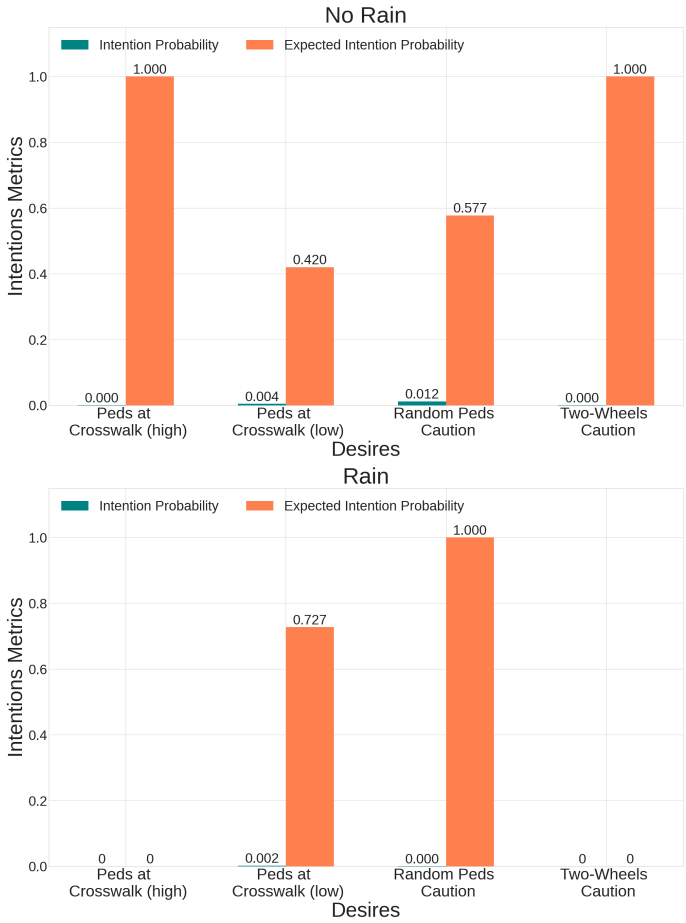


Figure 7.2: Comparison of driver intentions in rainy conditions versus clear conditions

that this intention will be fulfilled is lower compared to daytime scenarios. This suggests that drivers may allow pedestrians to pass on their side of the road without necessarily coming to a complete stop, or they may ignore pedestrians if they are crossing in the opposite direction. This behaviour may be influenced by the reduced presence of road elements and lower traffic levels at night compared to daytime conditions.

Furthermore, the vehicle's intention to slow done while approaching a traffic light is fulfilled more at night than during the day. A possible reason for this is that traffic lights typically display a blinking light at night, functioning similarly to a stop sign. Consequently, the colour of the traffic light does not affect the vehicle's behaviour, unlike during the day when the traffic light colour plays a significant role in driving decisions.

Similar to rainy conditions, intentions related to vulnerable road users are also less pronounced at night than in the daytime.

In addition to vulnerable road users, we identified another category of desires particularly affected by poor visibility conditions: stop sign desires. When analysing the intention metrics for stop sign-related desires, we observe that at night, unlike during the day, the vehicle never shows an intention to approach, stop or restart from stop signs. These metrics are also lower under rainy conditions compared to clear weather. This suggests the need for developers to doublecheck the vehicle's visual system's ability to accurately detect stop signs in challenging visibility conditions.

7.2.3 Unseen Conditions

To assess the generalisation capabilities of the PG under unseen yet plausible scenarios, in particular, those characterised by limited visibility, high visibility scenes were split into training and testing subsets using a 90/10 ratio. The PG is constructed on the training set (518 scenes) using discretiser D_{0b}, which proved to have the lowest entropy, thus reducing uncertainty in both action and future state predictions. The PG is tested on four sets: one derived from the split of high-

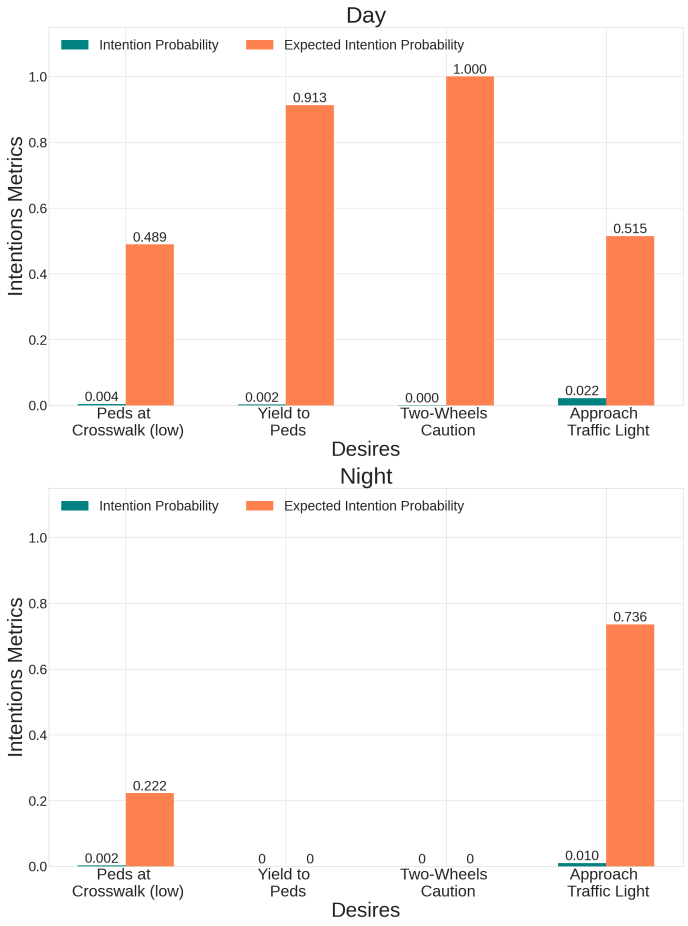


Figure 7.3: Comparison of driver intentions at daytime versus nighttime

visibility scenes (58 scenes) and three additional sets representing rainy scenes, nighttime scenes, and low-visibility conditions (a combination of night and rain).

We evaluate whether the PG can effectively generalise to adverse visibility conditions by computing the negative-log-likelihood (NLL) for each test set. The results of this analysis are presented in Table 7.1. The NLL values are generally consistent across different visibility conditions, suggesting that the PG maintains similar generalisation performance regardless of visibility levels. Interestingly, NLL values are slightly lower for scenes with worse visibility, which contrasts our expectations since the PG was constructed based on high-visibility scenes. A possible explanation for this finding is that high-visibility conditions can indeed lead to a wider variety of driving scenarios due to increased traffic volume and a wider range of road users.

Visibility	**N. Scenes**	$\overline{-l_a}$	$\overline{-l_w}$	$\overline{-l}$
Low (rain and night)	16	71.79	52.77	124.56
Medium (night)	99	76.11	51.43	127.54
Medium (rain)	171	68.40	63.62	132.02
High	58	77.82	54.55	132.37

Table 7.1: Neg-log-likelihood for different visibility conditions

Chapter 8

Conclusions

Research in Explainable Artificial Intelligence has gained significant importance as Artificial Intelligence is increasingly integrated into every aspect of life. This is especially relevant in developing autonomous vehicles, as these systems must operate in complex and unpredictable environments, where their decisions can have serious real-world consequences.

In this thesis, we studied *how* Policy Graphs and concepts from the Theory of Mind can be combined to explain the behaviour of an autonomous vehicle. The idea behind this approach is to replicate the way humans explain the behaviour of others by attributing desires and intentions. After reviewing current XAI techniques in the context of autonomous driving, we demonstrate that this method addresses several limitations of existing approaches. It offers global and local explanations of the behaviour of any AV model and provides proactive explanations by focusing on the agent's desires and intentions rather than just short-term context. The explanations are expressed in natural language predicates, ensuring they are interpretable for the explainee.

Our methodology begins with the sensor data collected from an AV and builds a graph-based representation of the agent's behaviour, integrating its hypothesised desires and intentions. This approach allows for extracting motivations behind specific driving decisions while also enabling the identification of unusual or undesirable behaviours. In particular, we can attribute an intention to the driver 81.2% of the time, which results in the fulfilment of the corresponding desire in 95.2% of cases, thereby providing an explanation for *almost* all vehicle decisions.

Secondly, we analyse the nuScenes dataset to identify potentially harmful *biases*. Our focus is mainly on class imbalances in the context of object detections, where vulnerable road users are underrepresented, and how these imbalances can affect the performance of the AV's visual detection system.

Thirdly, we investigate how diverse visibility conditions, such as bad weather and poor lighting, affect driving behaviour. This analysis reveals that intentions related to stop signs and vulnerable road users are rarely attributed to poor light and weather conditions. This underscores potential fallacies in the detection system towards these critical elements.

Finally, we show the generalisation capabilities of our methodology when confronted with unseen, plausible scenarios. The results show that a PG built from data collected in favourable conditions can also represent decision-making processes under poor visibility conditions.

The findings of this study can significantly benefit various stakeholders: it can assist automotive researchers in identifying vulnerabilities within these systems; help regulators verify whether the system operates within acceptable legal boundaries; and increase public trust through a deeper understanding of a vehicles' decisions.

8.1 Limitations and Future Research

This work presents two main limitations. First, the nuScenes dataset has several constraints. It includes only 5.5 hours of driving data, with approximately 15% lacking annotations, making those portions unusable for this study. Furthermore, nuScenes primarily focuses on basic driving scenarios and covers very few unusual situations (*e.g.*, scenes with ambulances or animals). A solution can be to incorporate the nuPlan dataset [50] from the same data provider. NuPlan offers a much larger dataset, including 1,500 hours of driving data, with more edge-case scenarios. Additionally, nuPlan provides dynamic map information, such as real-time traffic light statuses, which could further improve the robustness of our analyses.

The second limitation refers to our methodology, particularly the state discretisation process.

This step involves creating high-level, interpretable predicates that simplify complex state spaces into more meaningful and manageable representations. However, this discretisation process, together with the desire formulations, requires extensive domain knowledge.

A primary proposal for future improvements is to extend the desire formulation to comprehend more general and long-term desires. Examples of such desires could include minimising fuel consumption, arriving at a destination, reducing travel time, and the vehicle's environmental impact. As long-term goals motivate the agent's actions over an extended period, this would require longer driving scenes. A second possible improvement involves expanding the framework to account for multi-agent interactions. In this scenario, the ego vehicle's decisions would be influenced not only by its desires and intentions but also by the behaviour of other agents in its neighbourhood.

Bibliography

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. «nuScenes: A multimodal dataset for autonomous driving»». *CoRR*, abs/1903.11027, 2019.

[2] Anton Kuznietsov, Balint Gyevnar, Cheng Wang, Steven Peters, and Stefano V. Albrecht. «Explainable AI for Safe and Trustworthy Autonomous Driving: A Systematic Review», 2024.

[3] Oliver Willers, Sebastian Sudholt, Shervin Raafatnia, and Stephanie Abrecht. «Safety Concerns and Mitigation Approaches Regarding the Use of Deep Learning in Safety-Critical Perception Tasks», 2020.

[4] Brittany Moye. «AAA: Fear of Self-Driving cars on the rise», 2023. URL: https://newsroom.aaa.com/2023/03/aaa-fear-of-self-driving-cars-on-the-rise/. Accessed: September 20, 2024.

[5] Santokh Singh. «Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey». Technical report, NHTSA's National Center for Statistics and Analysis, 2018.

[6] Neville A. Stanton, Paul M. Salmon, Guy H. Walker, and Maggie Stanton. «Models and methods for collision analysis: A comparison study based on the Uber collision with a pedestrian». *Safety Science*, volume 120, pages 120-123, 2019.

[7] Tim Miller. «Explanation in Artificial Intelligence: Insights from the Social Sciences», 2018. pages 3,4,14.

[8] Bradley Hayes and Julie A. Shah. «Improving Robot Controller Transparency Through Autonomous Policy Explanation». In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 303–312, 2017.

[9] Victor Gimenez-Abalos, Sergio Alvarez-Napagao, Adrián Tormos, Ulises Cortés, and Javier Vazquez-Salceda. «Intention-aware policy graphs: answering what, how, and why in opaque agents». 2024. Chapter 4, pages 13-33.

[10] Victor Gimenez-Abalos, Luis Oliva-Felipe, Javier Vázquez-Salceda, Ulises Cortés, and Sergio Álvarez Napagao. «Why Interpreting Intent Is Key for Trustworthiness in the Age of Opaque Agents», 2024.

[11] Victor Gimenez-Abalos. «Toward Explainable Agent Behaviour». In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, page 2740–2742, Richland, SC, 2024. International Foundation for Autonomous Agents and Multiagent Systems.

[12] Marc Domenech i Vila, Dmitry Gnatyshak, Adrian Tormos, Victor Gimenez-Abalos, and Sergio Alvarez-Napagao. «Explaining the Behaviour of Reinforcement Learning Agents in a Multi-Agent Cooperative Environment Using Policy Graphs». *Electronics*, 13(3), 2024.

[13] The Editors of Encyclopaedia Britannica. «Explanation. Encyclopedia Britannica.», 2017. URL: https://www.britannica.com/topic/explanation. Accessed: September 20, 2024.

[14] H. P. Grice. *«Logic and conversation»*. Academic Press, New York, 1975. Pages 43-47.

[15] Cynthia Rudin. «Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead», 2019. Pages 1-2.

[16] Finale Doshi-Velez and Been Kim. «Towards A Rigorous Science of Interpretable Machine Learning», 2017. Pages 3-4.

[17] Amina Adadi and Mohammed Berrada. «Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)». *IEEE Access*, 6:52138–52160, 2018.

[18] European Parliament and Council of the European Union. «General Data Protection Regulation (EU Regulation 2016/679)», 2016.

[19] European Parliament and Council of the European Union. «Artificial Intelligence Act (Regulation (EU) 2024/1689), Official Journal version of 13 June 2024». Interinstitutional File: 2021/0106(COD), 2024.

[20] Timo Speith. «A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods». pages 2239–2250, 2022.

[21] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius Brito Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago Paixão, Filipe Mutz, Lucas Veronese, Thiago Oliveira-Santos, and Alberto Ferreira De Souza. «Self-Driving Cars: A Survey», 2019.

[22] Shahin Atakishiyev, Mohammad Salameh, Hengshuai Yao, and Randy Goebel. «Explainable Artificial Intelligence for Autonomous Driving: A Comprehensive Overview and Field Guide for Future Research Directions», 2024.

[23] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. «Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization». *International Journal of Computer Vision*, 128(2):336–359, 2019.

[24] Suresh Kolekar, Shilpa Gite, Biswajeet Pradhan, and Abdullah Alamri. «Explainable AI in Scene Understanding for Autonomous Vehicles in Unstructured Traffic Environments on Indian Roads Using the Inception U-Net Model with Grad-CAM Visualization». *Sensors*, 22(24), 2022.

[25] Harsh Mankodiya, Dhairya Jadav, Rajesh Gupta, Sudeep Tanwar, Wei-Chiang Hong, and Ravi Sharma. «OD-XAI: Explainable AI-Based Semantic Object Detection for Autonomous Vehicles». *Applied Sciences*, 12(11), 2022.

[26] Amirata Ghorbani, Abubakar Abid, and James Zou. «Interpretation of Neural Networks is Fragile», 2018.

[27] Scott Lundberg and Su-In Lee. «A Unified Approach to Interpreting Model Predictions», 2017.

[28] Zhihao Cui, Meng Li, Yanjun Huang, Yulei Wang, and Hong Chen. «An interpretation framework for autonomous vehicles decision-making via SHAP and RF». In *2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, pages 1–7, 2022.

[29] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. «Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods», 2020.

[30] Guillaume Chaslot, Sander Bakkes, Istvan Szita, and Pieter Spronck. «Monte-Carlo Tree Search: A New Framework for Game AI». 2008.

[31] Balint Gyevnar, Massimiliano Tamborski, Cheng Wang, Christopher G. Lucas, Shay B. Cohen, and Stefano V. Albrecht. «A Human-Centric Method for Generating Causal Explanations in Natural Language for Autonomous Vehicle Motion Planning». 2022.

[32] Xin-She Yang. «2 - Mathematical foundations». In Xin-She Yang, editor, *Introduction to Algorithms for Data Mining and Machine Learning*, pages 19–43. Academic Press, 2019.

[33] Balint Gyevnar, Cheng Wang, Christopher G. Lucas, Shay B. Cohen, and Stefano V. Albrecht. «Causal Explanations for Sequential Decision-Making in Multi-Agent Systems», 2024.

[34] Alex Campolo, Madelyn Sanfilippo, Meredith Whittaker, and Kate Crawford. «AI Now 2017 Report ». In *AI Now 2017 Symposium and Workshop*, 2018.

[35] Thomas Hellström, Virginia Dignum, and Suna Bensch. «Bias in Machine Learning What is it Good (and Bad) for?». *CoRR*, abs/2004.00686, 2020.

[36] O. Lam, B. Broderick, S. Wojcik, and A. Hughes. «Gender and Jobs in Online Image Searches». *Pew Social Trends*, 2018. URL: https://www.pewresearch.org/social-trends/2018/12/17/gender-and-jobs-in-online-image-searches/. Accessed: September 23, 2024.

[37] Vaibhav Kumar, Tenzin Singhay Bhotia, and Tanmoy Chakraborty. «Nurse is Closer to Woman than Surgeon? Mitigating Gender-Biased Proximities in Word Embeddings». *CoRR*, abs/2006.01938, 2020.

[38] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. «A Survey on Bias and Fairness in Machine Learning», 2022.

[39] Benjamin Wilson, Judy Hoffman, and Jamie Morgenstern. «Predictive Inequity in Object Detection», 2019.

[40] Sam Biddle. «Cruise knew its self-driving cars had problems recognizing children - and kept them on the streets». *The Intercept*, 2023. URL: https://theintercept.com/2023/11/06/cruise-self-driving-cars-children/. Accessed: September 23, 2024.

[41] David Danks and Alex London. «Algorithmic Bias in Autonomous Systems». pages 4691–4697, 2017.

[42] Deloitte. «Striving for fairness in AI models», 2022. Accessed: September 23, 2024.

[43] Sergio Alvarez-Napagao, Adrián Tormos, Victor Abalos, and Dmitry Gnatyshak. «Policy graphs in action: explaining single- and multi-agent behaviour using predicates». In *XAI in Action: Past, Present, and Future Applications*, 2023.

[44] Bertram Malle. «Folk Theory of Mind: Conceptual Foundations of Human Social Cognition». *The New Unconscious*, 2012.

[45] David Premack and Guy Woodruff. Does a chimpanzee have a theory of mind». *Behavioral and Brain Sciences*, 1:515 – 526, 1978.

[46] Tongtong Liu, Joe McCalmon, Thai Le, Dongwon Lee, and Sarra Alqahtani. «A Policy-Graph Approach to Explain Reinforcement Learning Agents: A Novel Policy-Graph Approach with Natural Language and Counterfactual Abstractions for Explaining Reinforcement Learning Agents», 2022.

[47] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. «DriveLM: Driving with Graph Visual Question Answering», 2023.

[48] Nassim Belmecheri, Arnaud Gotlieb, Nadjib Lazaar, and Helge Spieker. «Towards Trustworthy Automated Driving through Qualitative Scene Understanding and Explanations», 2024.

[49] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M. Alvarez. «Is Ego Status All You Need for Open-Loop End-to-End Autonomous Driving?», 2023.

[50] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. «NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles», 2022.

Appendix A

Early Implementation Stage Metrics

We present desire and intention metrics for each desire across all discretisations, as observed during the early stage of the work.

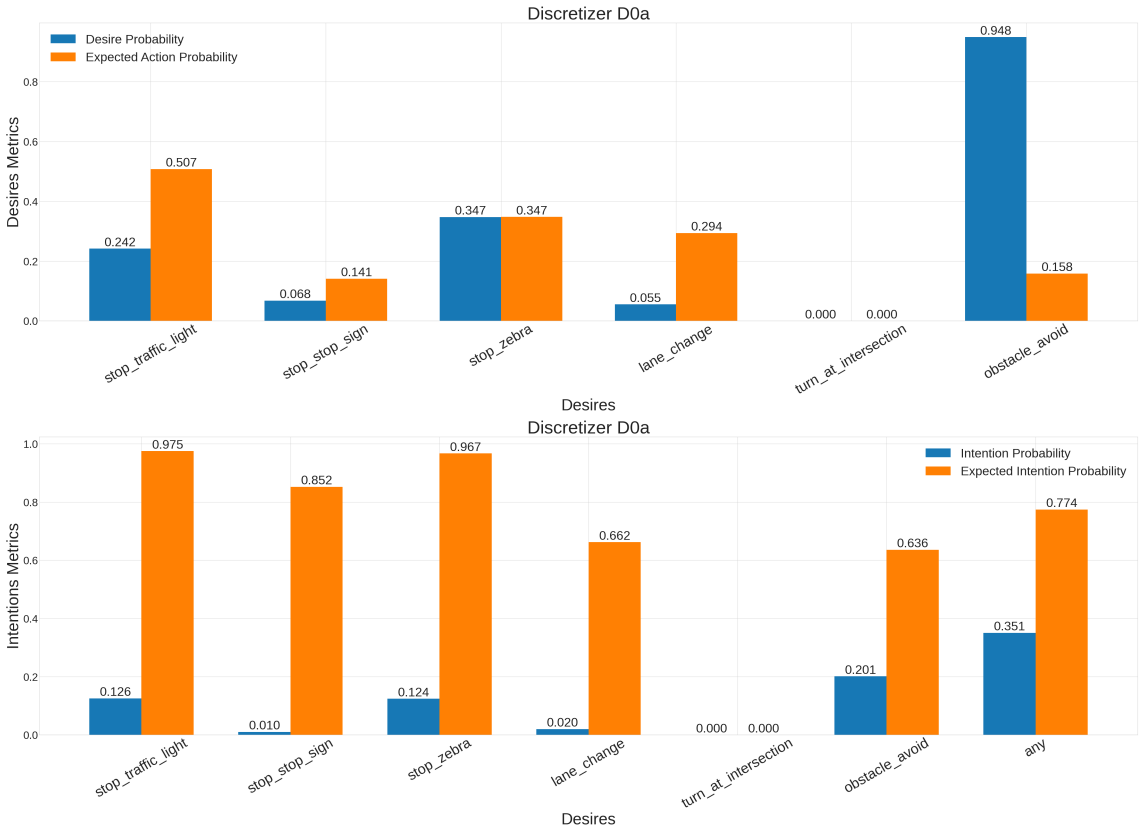


Figure A.1: Desire and intention metrics for D_{0a}

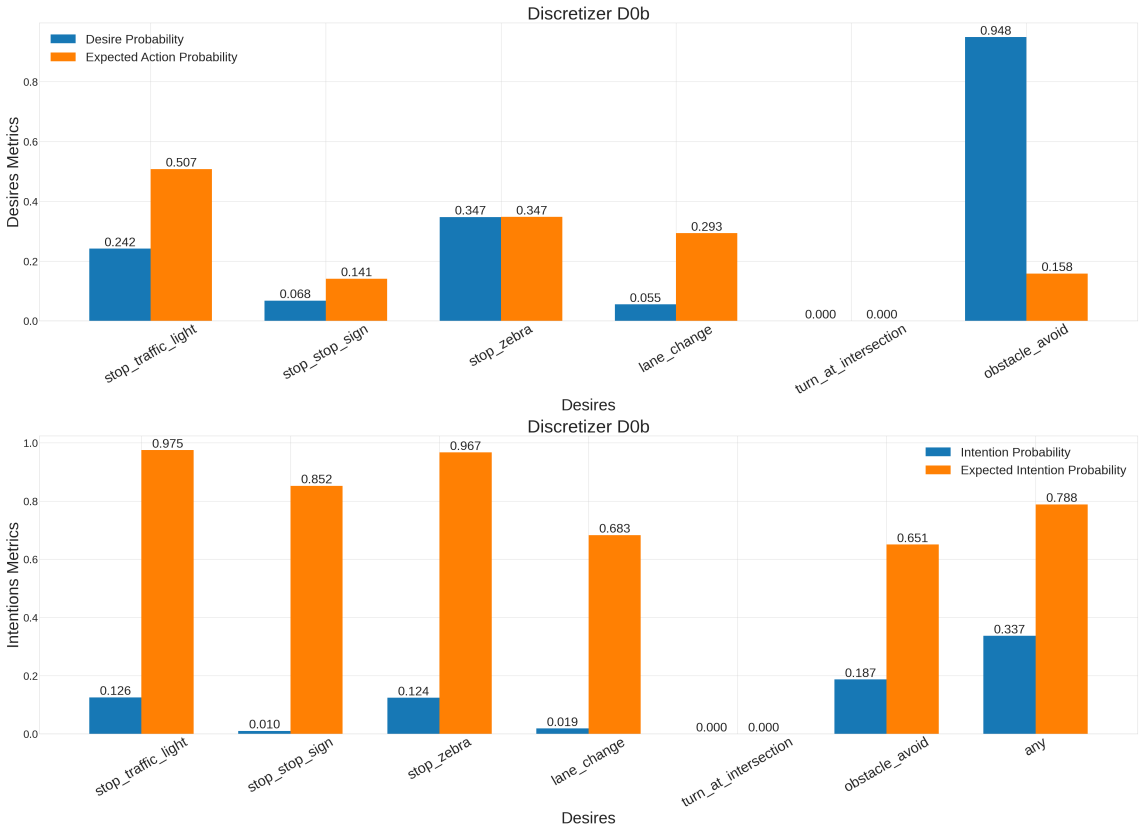


Figure A.2: Desire and intention metrics for D_{0b}

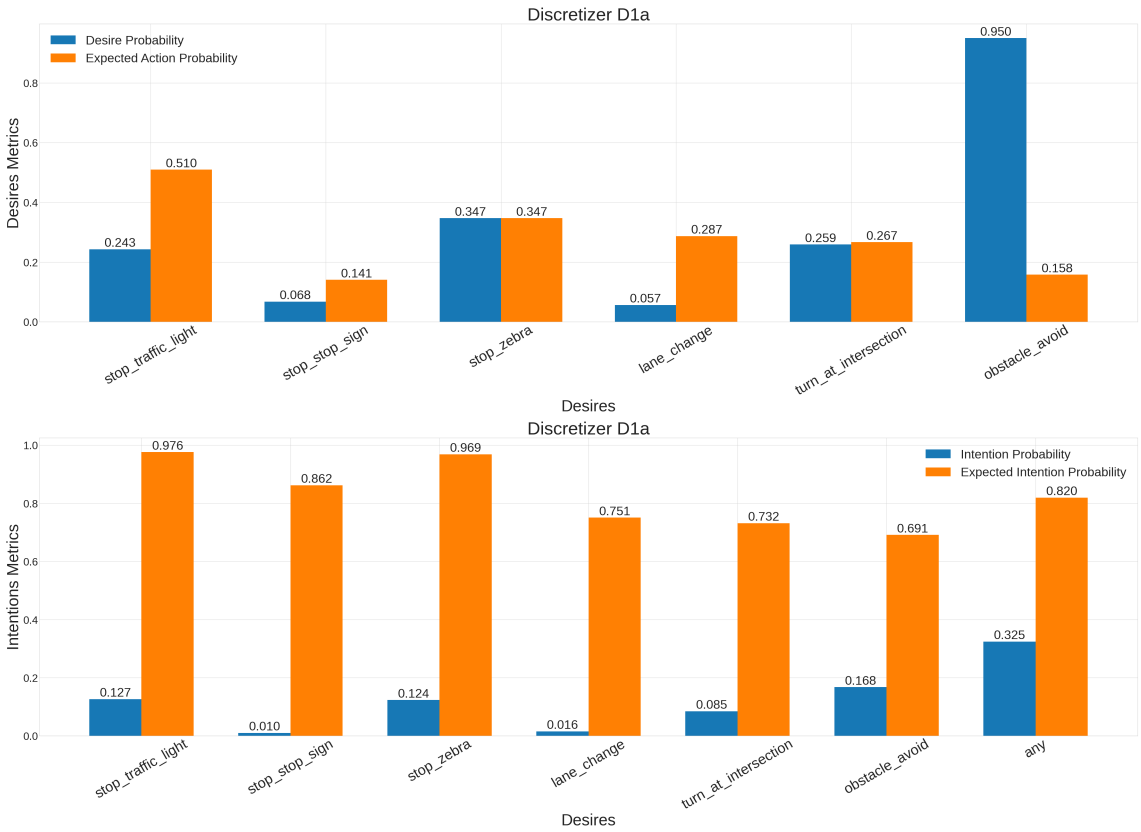


Figure A.3: Desire and intention metrics for D_{1a}

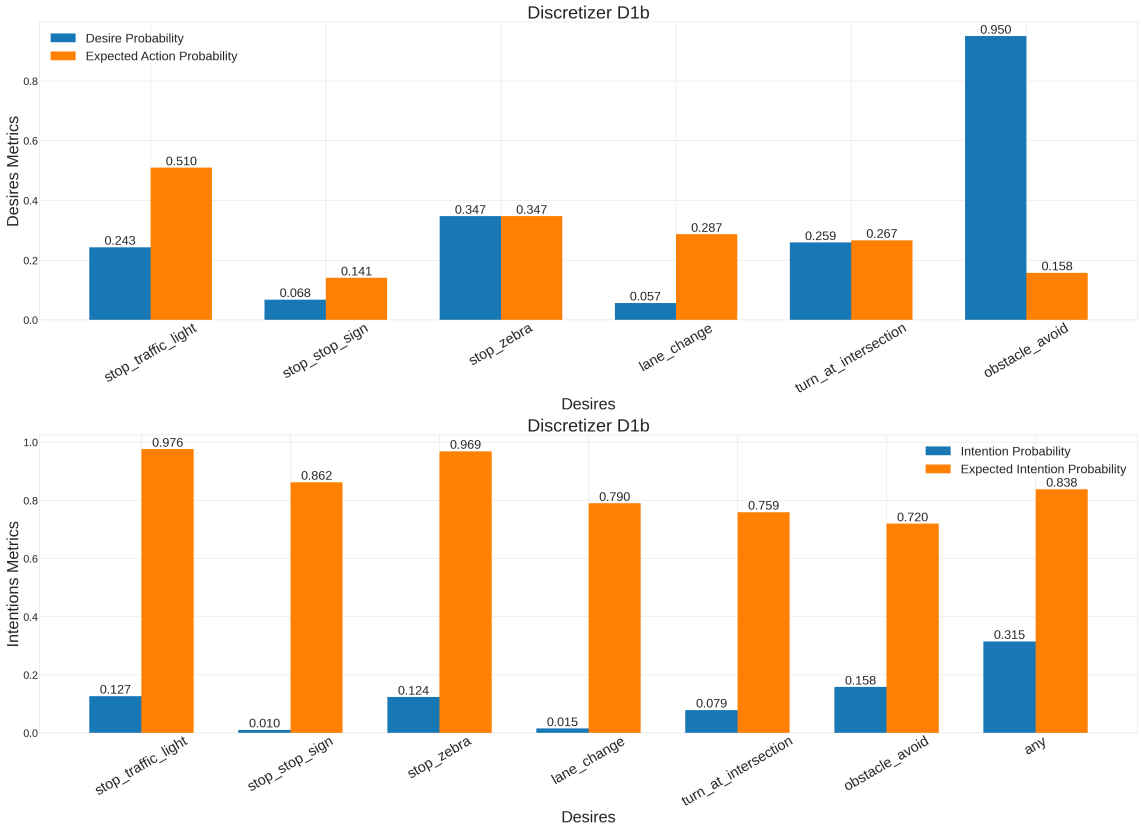


Figure A.4: Desire and intention metrics for D_{1b}

Appendix B

Final Implementation Stage Metrics

We present desire and intention metrics for each formalised desire across all discretisations, as observed during the final stage of the work.

B.1 Desire Metrics

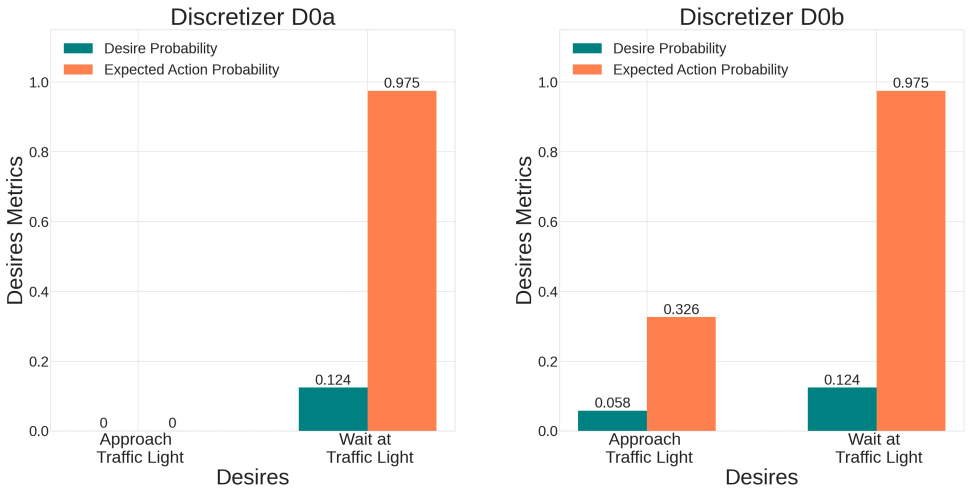


Figure B.1: Desire metrics for *traffic lights* desires for D_{0a} and D_{0b}. Results for D_{2a} and D_{1a} are omitted as they are identical to D_{0a}, and results for D_{2b} and D_{2b} are omitted as they are identical to D_{0b}.

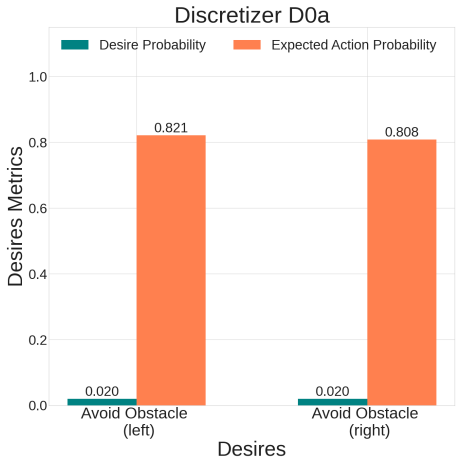


Figure B.2: Desire metrics for *obstacle avoidance* desires for D_{0a}. The results are identical across all discretisers.

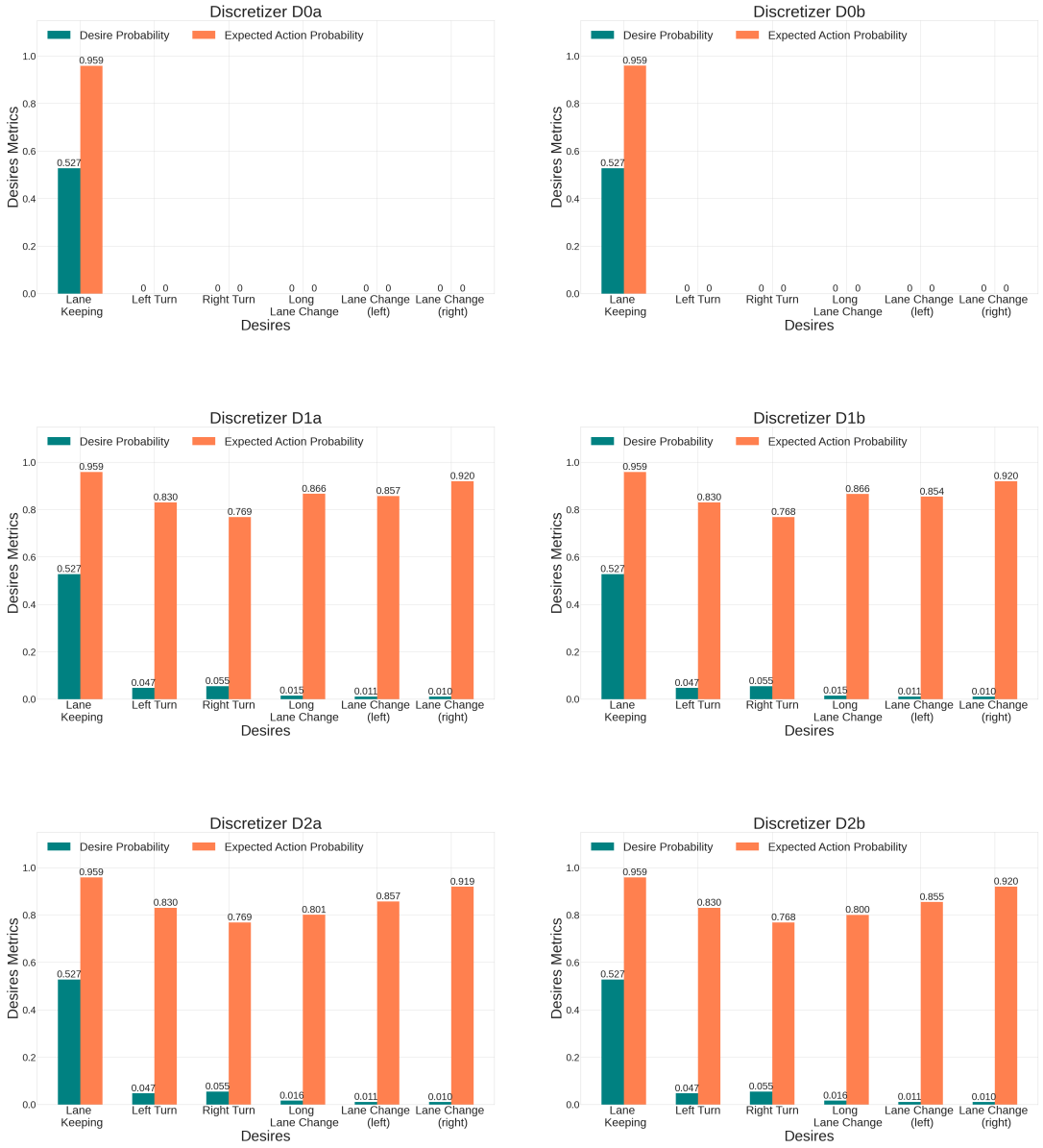


Figure B.3: Desire metrics for *cruising* desires

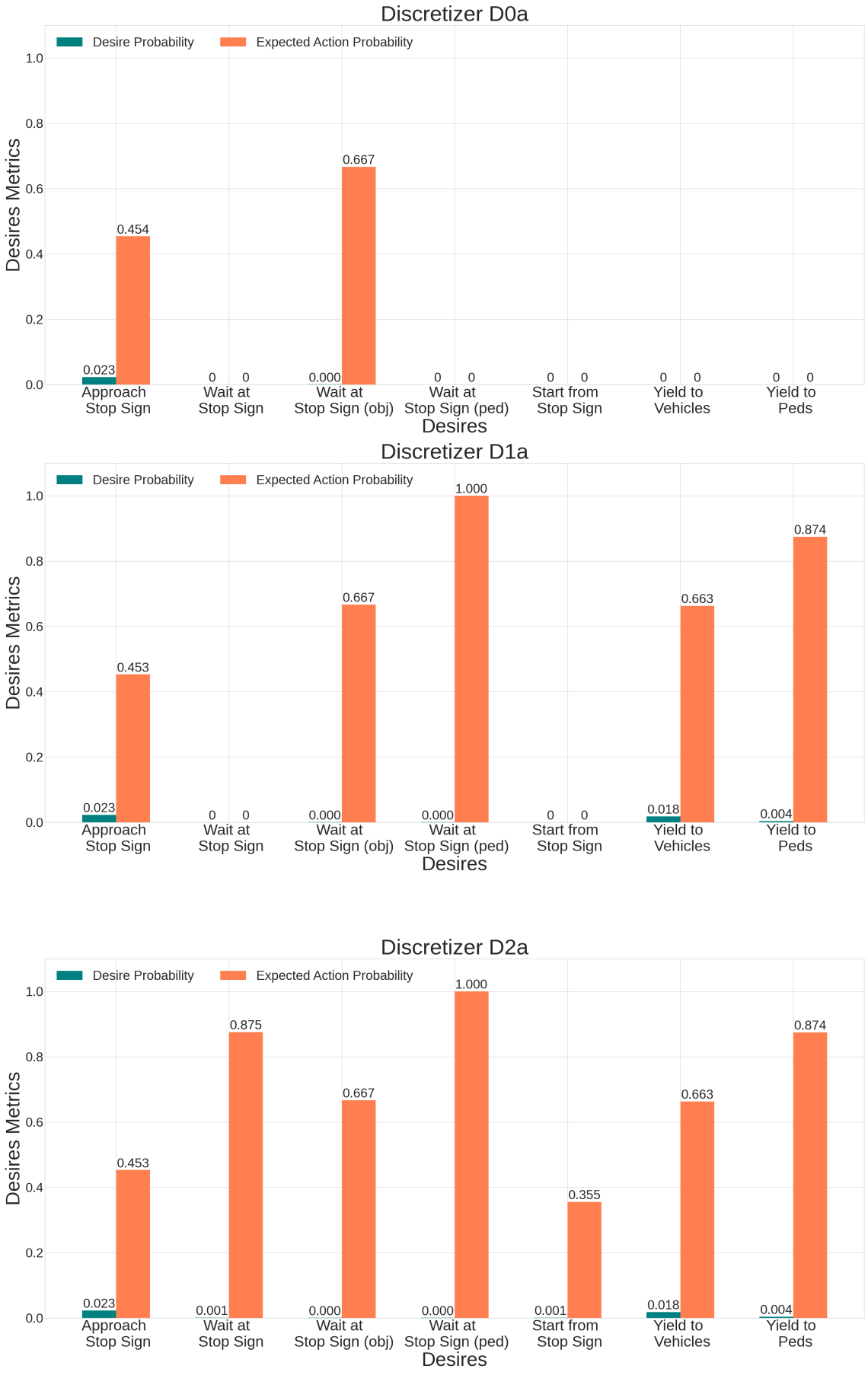


Figure B.4: Desire metrics for *stop area* desires for D_{0a}, D_{1a} and D_{2a}. Results for discretisers of type b are omitted as they the same as for their corresponding a version.

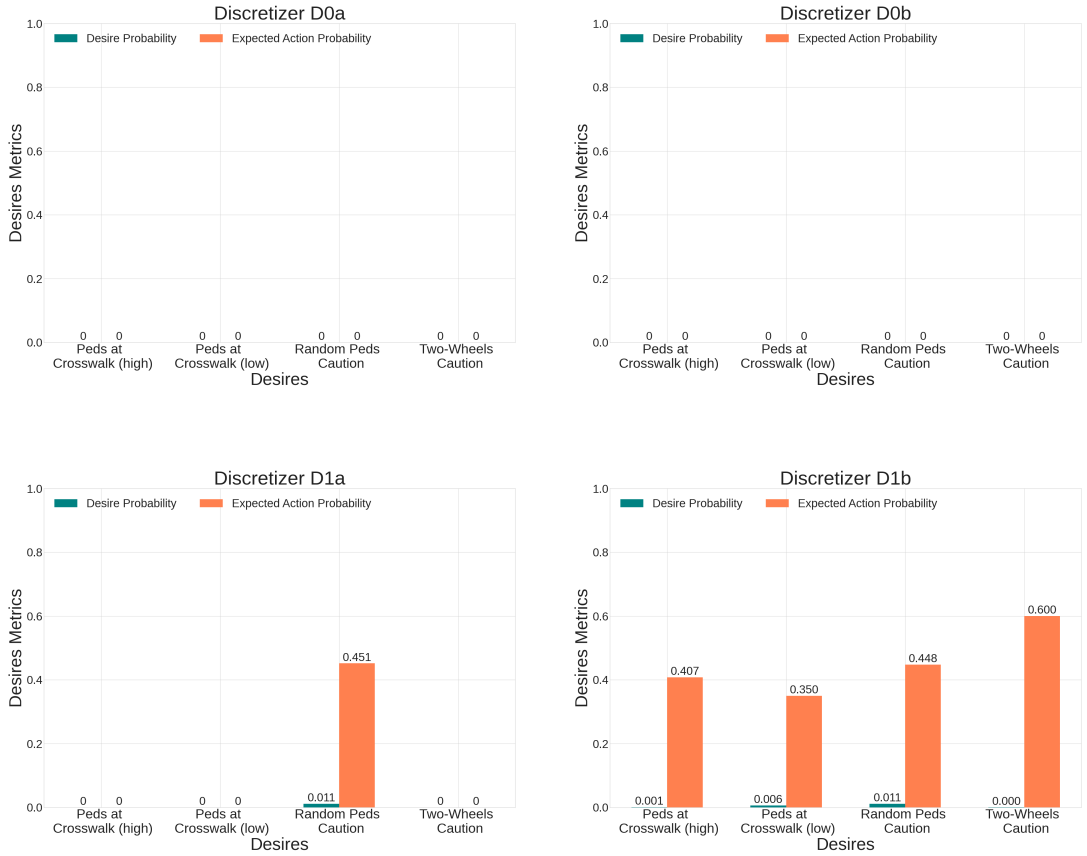


Figure B.5: Desire metrics for *vulnerable road users* desires. Results for D_{2a} and D_{2b} are omitted since they are the same as for D_{1a} and D_{1b}.

B.2 Intention Metrics

Figure B.6: Desire metrics for *unsafe* desires. Results for D_{2a} and D_{2b} are omitted as they are the same as for D_{1a} and D_{1b} respectively.

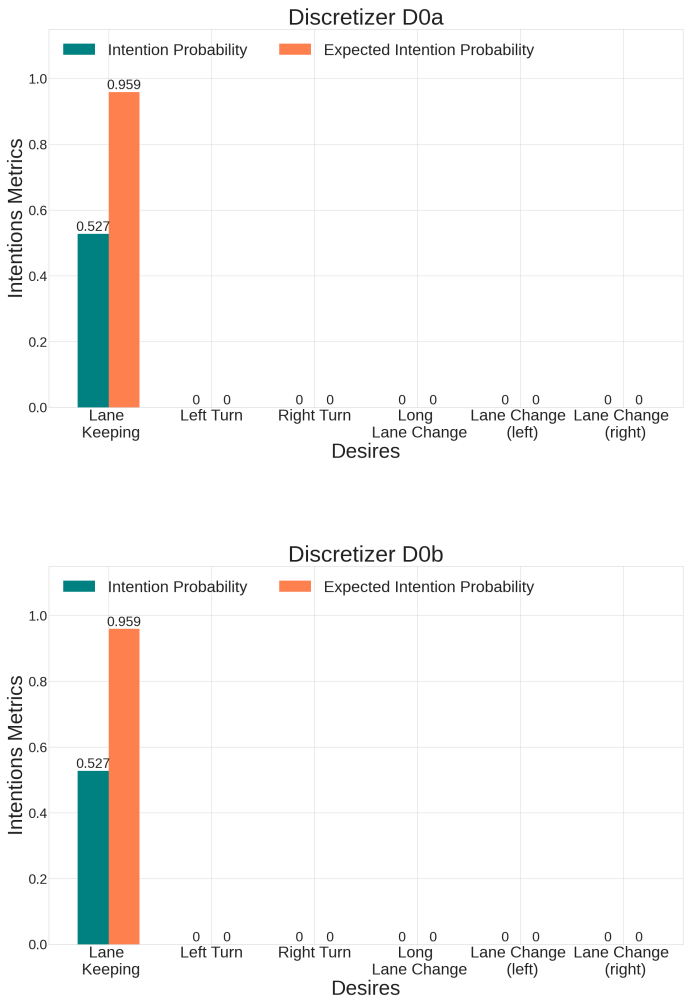


Figure B.7: Intention metrics for *cruising* desires. Results for D_{2a} and D_{1a} are omitted as they are the same as for D_{0a}. D_{2b} and D_{1b} are omitted as they are the same as for D_{0b}.

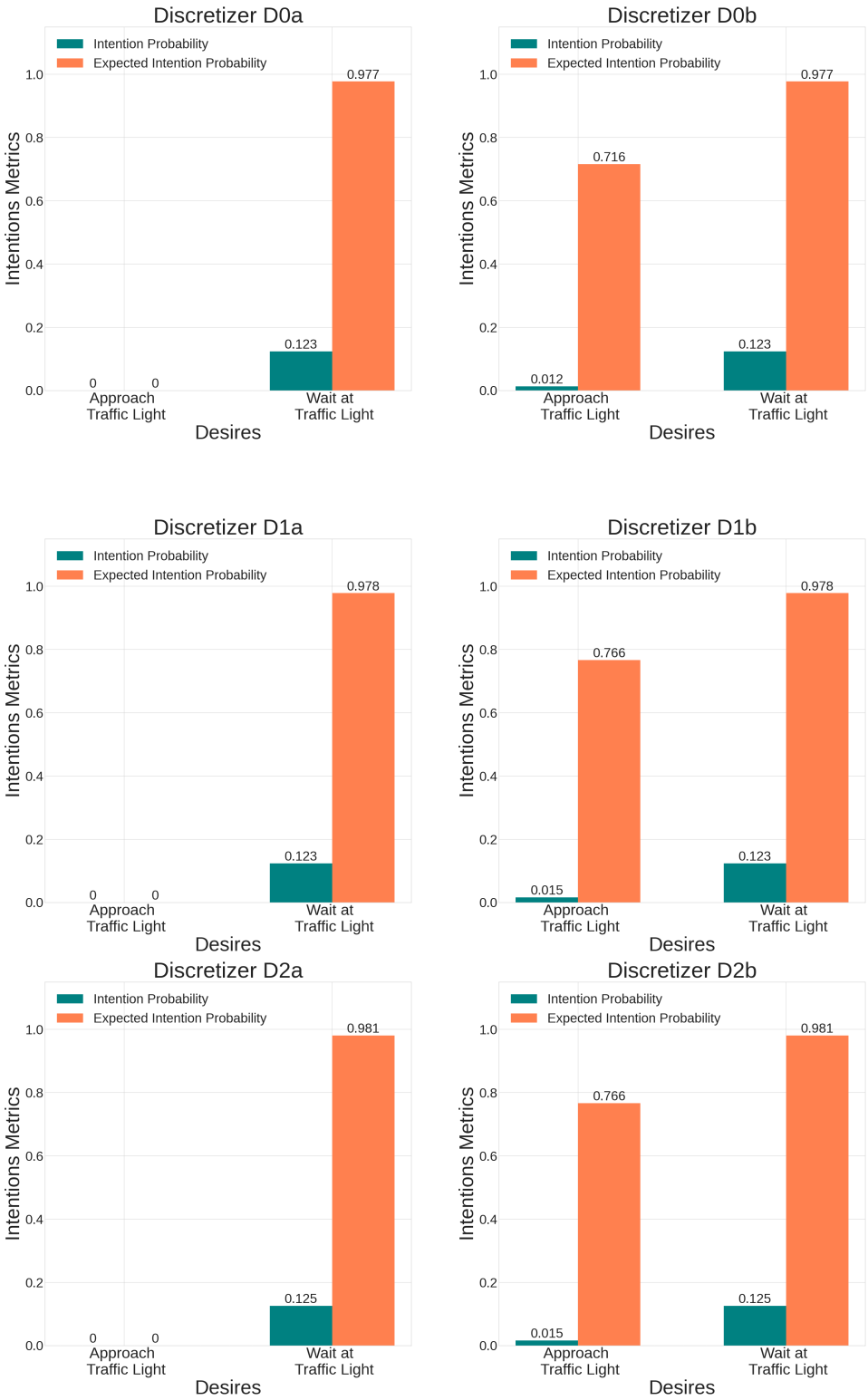


Figure B.8: Intention metrics for near *traffic lights* desires. Results for D_{2a} and D_{2b} are omitted as they are the same as for D_{1a} and D_{1b} respectively.

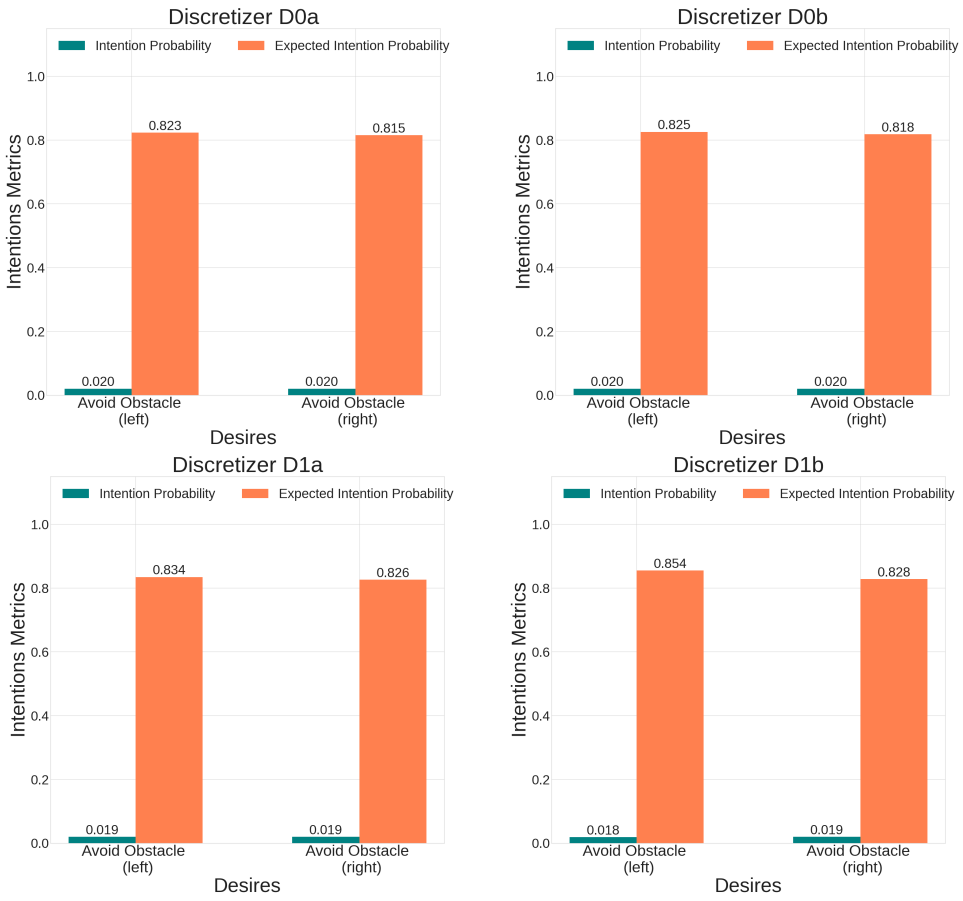
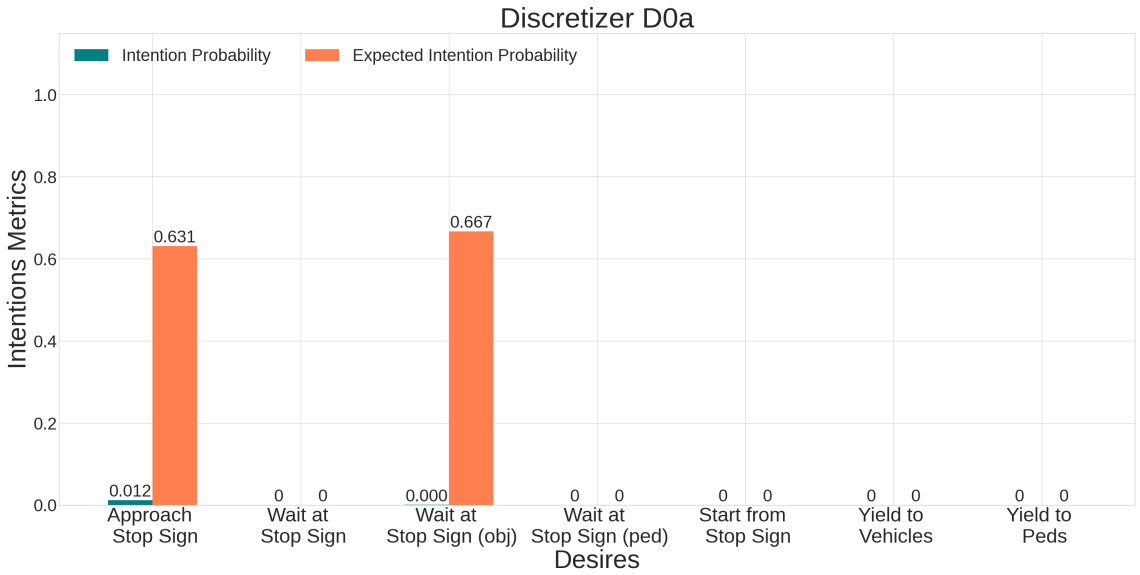
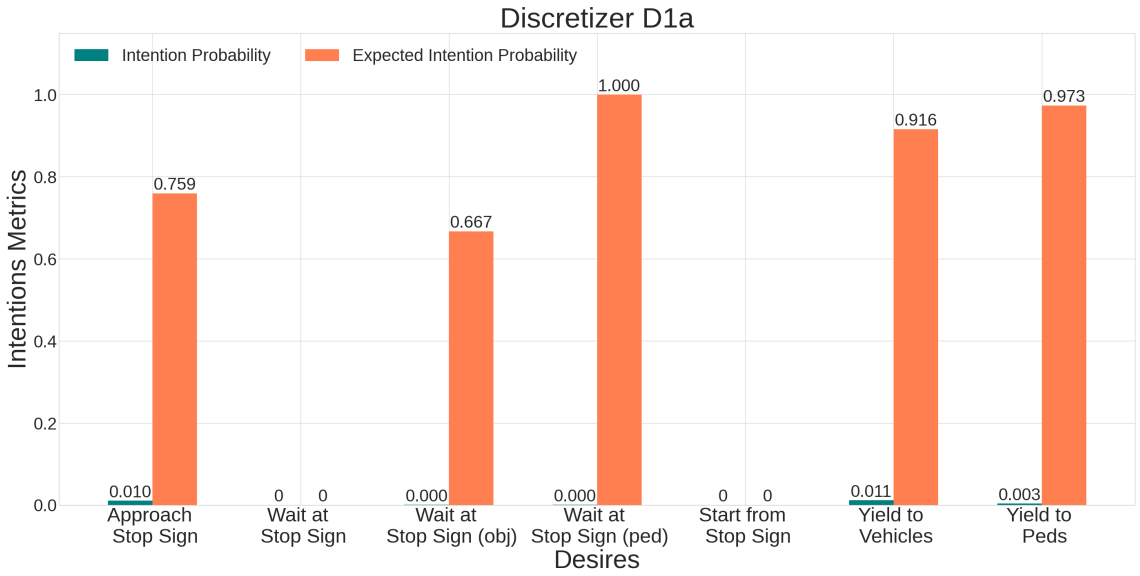


Figure B.9: Intention metrics for *obstacle avoidance* desires. Results for D_{2a} and D_{2b} are omitted as they are the same as for D_{1a} and D_{1b} respectively.





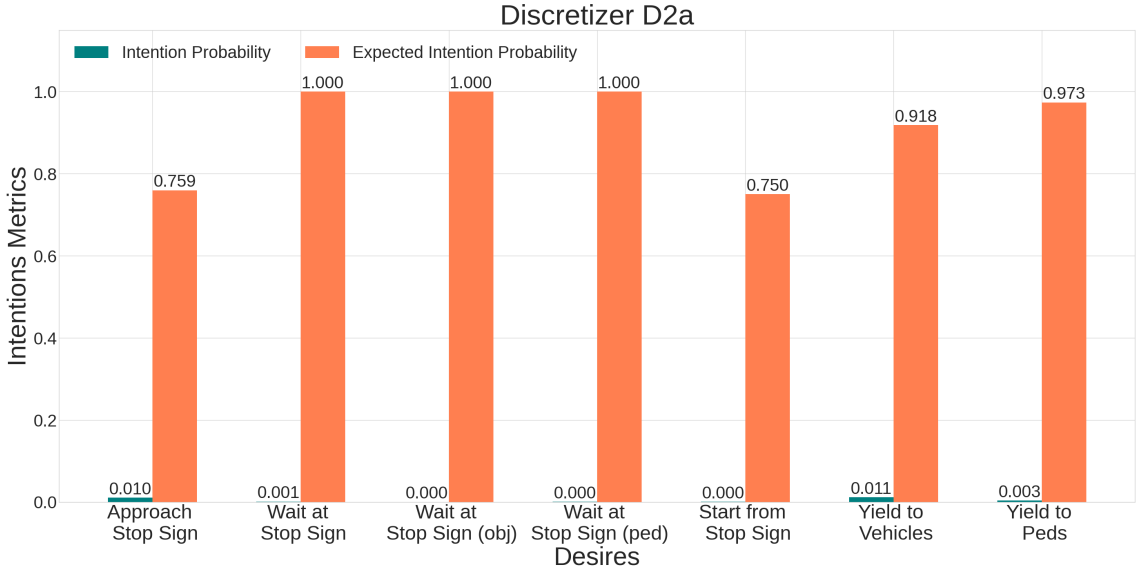


Figure B.10: Intention metrics for *stop area* desires for D_{0a}, D_{1a} and D_{2a}. Results for discretisers of type b are omitted as they the same as for their corresponding a version.

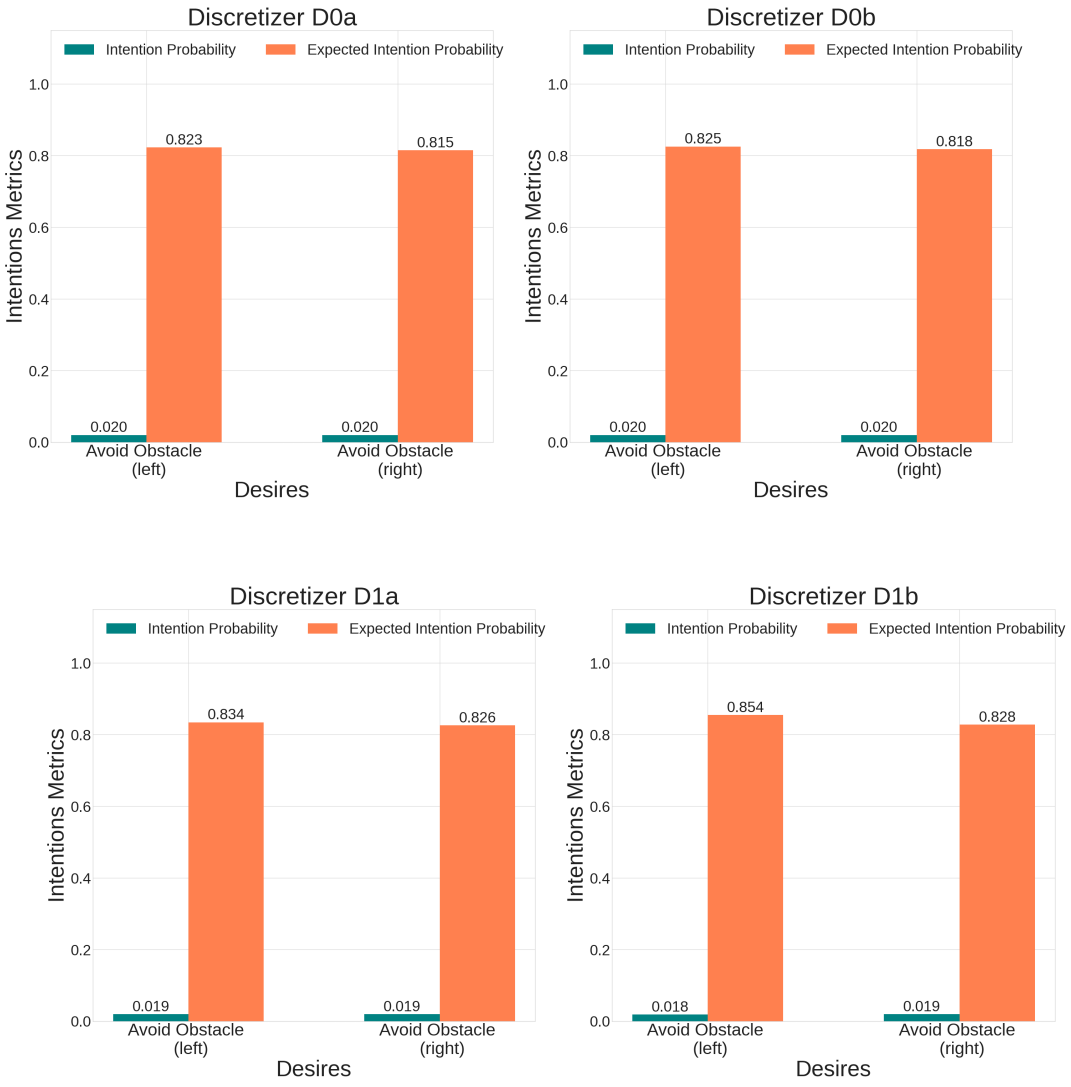


Figure B.11: Intention metrics for *vulnerable road users* desires. Results for D_{2a} and D_{2b} are omitted as they are the same as for D_{1a} and D_{1b} respectively.

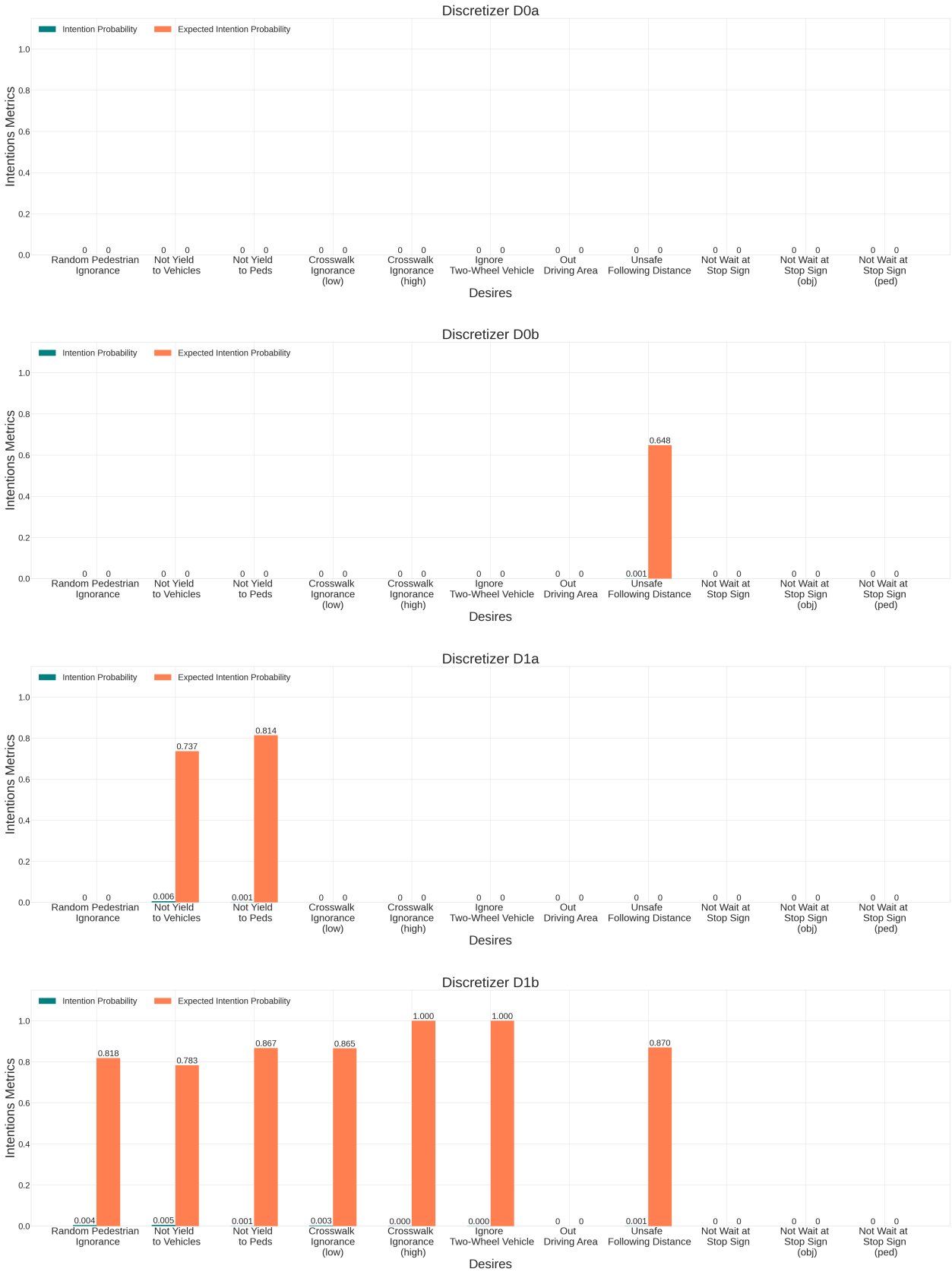


Figure B.12: Intention metrics for *unsafe* desires. Results for D_{2a} and D_{2b} are omitted as they are the same as for D_{1a} and D_{1b} respectively.