Learning Multi-Modal Self-Awareness Models Empowered by Active Inference for Autonomous Vehicles

Universidad
Carlos III de Madrid

Sheida Nozari

Department of Electrical, Electronic, Telecommunications Engineering and
Naval Architecture (DITEN)
University of Genoa

Department of Systems Engineering and Automation (DISA)
University Carlos III of Madrid

This dissertation is submitted for the degree of
Doctor of Philosophy

Joint Doctorate in Interactive and
Cognitive Environments - Cycle 35

April 2023

Learning Multi-Modal Self-Awareness Models Empowered by Active Inference for Autonomous Vehicles

Sheida Nozari

Joint Doctorate in Interactive and Cognitive Environments
JD-ICE

XXXV cycle

Acknowledgements

This PhD Thesis has been developed in the framework of, and according to, the rules of the Joint Doctorate in Interactive and Cognitive Environments JD-ICE with the cooperation of the following Universities:

Università degli Studi di Genova (UNIGE)

DITEN - Dept. of Electrical, Electronic, Telecommunications Engineering and Naval Architecture

ISIP40 - Information and Signal Processing for Cognitive Telecommunications

Primary Supervisor: Prof. Carlo Regazzoni
Co-Supervisor: Prof. Lucio Marcenaro

Universidad Carlos III de Madrid (UC3M)

IEEA - Doctoral Program in Electrical Engineering, Electronics and Automation

DISA - Dep. of Systems Engineering and Automation

ISL - Intelligent System Laboratory
Supervisor: Prof. David Martín Gomez

Universidad
Carlos III de Madrid

"O snail
little by little
climb Mount Fuji"

-Kobayashi Issa

Acknowledgements

First and foremost I am extremely grateful to my supervisors. The completion of this Ph.D. would not have been possible without the guidance and support of my advisors. I would like to express my deepest gratitude to Prof. Carlo Regazzoni for his wise guidance, patience, and continuous support of my Ph.D. research. My sincere gratitude also goes to Prof. David Martín Gomez for his guidance and encouragement. I would also have to thank Prof. Lucio Marcenaro for his priceless support, availability, and advice. To end this, I appreciate my supervisors and mentors from whom I learned the crucial aspects of being a good researcher.

I thank my fellow labmates at the University of Genoa for the stimulating discussions, the sleepless nights we were working together getting close to deadlines, and all the fun times we have had in the last three years. In particular, I thank my colleagues, or better, my friends Mohamad Baydoun and Damian Campo who smoothed the first year of my Ph.D. I thank Hafsa Iqbal who listened carefully and was always there to extend a helping hand. My cordial thanks go to Ali Krayani, without his support I did not have an easy time in the course of the last year. Nor can I forget naming Giulia Slavic, Abraham Shiferaw Alemaw, Felix Obite, Nobel John Willian, Khalid Khan, and Muhammad Adnan for their loving support. Also, I thank my fellow at the University Carlos III of Madrid, Pablo Marín Plaza who is a teacher inherently.

My sincere thanks to all my friends who lovingly accompanied me on the important phases of this journey of exploration.

Last but not least, my family has been an essential part of the great chain of support I have had throughout my life and career. My heartfelt gratitude for what they have done for me. In particular, David Reversat whom I did not let sleep by nerve-breaking clicks on the mouse many nights, I promise I will change my mouse.

Published and Submitted Content

0.1 Journal Papers

1. **S. Nozari**, A. Krayani, P. Marin-Plaza, L. Marcenaro, D. M. Gómez and C. Regazzoni, "Exploring Action-Oriented Models via Active Inference for Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2023 (Under Review).

   (This journal paper is wholly included in the thesis, chapter 4, and chapter 5. The material from this source included in this thesis is not singled out with typographic means and references.)

2. **S. Nozari**, A. Krayani, P. Marin-Plaza, L. Marcenaro, D. M. Gómez and C. Regazzoni, "Active Inference Integrated With Imitation Learning for Autonomous Driving," *IEEE Access*, vol. 10, pp. 49738-49756, 2022, doi: 10.1109/ACCESS.2022.3172712.

   (This journal paper is wholly included in the thesis, chapter 3, and chapter 4. The material from this source included in this thesis is not singled out with typographic means and references.)

0.2 Conference Papers

1. **S. Nozari**, A. Krayani, P. Marin-Plaza, L. Marcenaro, D. M. Gómez and C. Regazzoni, "Adapting Exploratory Behaviour in Active Inference for Autonomous Driving," *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2023*. (Accepted to appear).

   (This conference paper is wholly included in the thesis, chapter 4. The material from this source included in this thesis is not singled out with typographic means and references.)

2. **S. Nozari**, A. Krayani, P. Marin-Plaza, L. Marcenaro, D. M. Gómez and C. Regazzoni, "Autonomous Driving Based on Imitation and Active Inference," *2023 Advances*

in System-Integrated Intelligence (SYSINT), 2023, vol 546. Springer, Cham. doi: 10.1007/978-3-031-16281-7-2.

(This conference paper is partly included in the thesis, chapter 3. The material from this source included in this thesis is not singled out with typographic means and references.)

3. **S. Nozari**, A. Krayani, L. Marcenaro, D. Martin and C. Regazzoni, "Incremental Learning through Probabilistic Behavior Prediction," *2022 30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 1502-1506, doi: 10.23919/EUSIPCO55093.2022.9909735.

   (This conference paper is wholly included in the thesis, chapter 3. The material from this source included in this thesis is not singled out with typographic means and references.)

4. **S. Nozari**, L. Marcenaro, D. Martin and C. Regazzoni, "Observational Learning: Imitation Through an Adaptive Probabilistic Approach," *2021 IEEE International Conference on Autonomous Systems (ICAS)*, 2021, pp. 1-5, doi: 10.1109/ICAS49788.2021.9551152.

   (This conference paper is wholly included in the thesis, chapter 3. The material from this source included in this thesis is not singled out with typographic means and references.)

Abstract

For autonomous agents to coexist with the real world, it is essential to anticipate the dynamics and interactions in their surroundings. Autonomous agents can use models of the human brain to learn about responding to the actions of other participants in the environment and proactively coordinates with the dynamics. Modeling brain learning procedures is challenging for multiple reasons, such as stochasticity, multi-modality, and unobservant intents. A neglected problem has long been understanding and processing environmental perception data from the multisensorial information referring to the cognitive psychology level of the human brain process. The key to solving this problem is to construct a computing model with selective attention and self-learning ability for autonomous driving, which is supposed to possess the mechanism of memorizing, inferring, and experiential updating, enabling it to cope with the changes in an external world. Therefore, a practical self-driving approach should be open to more than just the traditional computing structure of perception, planning, decision-making, and control. It is necessary to explore a probabilistic framework that goes along with human brain attention, reasoning, learning, and decision-making mechanism concerning interactive behavior and build an intelligent system inspired by biological intelligence.

This thesis presents a multi-modal self-awareness module for autonomous driving systems. The techniques proposed in this research are evaluated on their ability to model proper driving behavior in dynamic environments, which is vital in autonomous driving for both action planning and safe navigation. First, this thesis adapts generative incremental learning to the problem of imitation learning. It extends the imitation learning framework to work in the multi-agent setting where observations gathered from multiple agents are used to inform the training process of a learning agent, which tracks a dynamic target. Since driving has associated rules, the second part of this thesis introduces a method to provide optimal knowledge to the imitation learning agent through an active inference approach. Active inference is the selective information method gathering during prediction to increase a predictive machine learning model's prediction performance. Finally, to address the inference complexity and solve the exploration-exploitation dilemma in unobserved environments, an

exploring action-oriented model is introduced by pulling together imitation learning and active inference methods inspired by the brain learning procedure.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

3-D	three-dimensional
AD	Autonomous Driving
ADS	Autonomous Driving System
AFP-M	Active First-Person Model
AIn	Active Inference
AS	Autonomous System
AV	Autonomous Vehicle
BN	Bayesian Network
C-GDBN	Coupled Generalized Dynamic Bayesian Network
CL	Continuous Level
CS	Cognitive Science
DBN	Dynamic Bayesian Network
DL	Discrete Level
E	Expert
EFE	Expected Free Energy
FE	Free Energy
FP-M	First-Person Model

GDBN Generalized Dynamic Bayesian Network

GE Generalized Error

GM Generative Model

GNG Growing Neural Gas

GS Generalized State

IA Intelligent Agent

IL Imitation Learning

IRL Inverse Reinforcement Learning

ITS Intelligent transportation system

KF Kalman Filter

KL Kullback Leibler-Divergence

L Learner

MDP Markov Decision Process

MJPF Markov Jump Particle Filter

ML Machine Learning

NFF Null Force Filter

PF Particle Filter

PGM Probabilistic Graphical Model

RL Reinforcement Learning

RM Reference Model

SA Self-Awareness

SL Self Learning

SM Situation Model

VFE Variational Free Energy

WM World Model

Chapter 1

Introduction

1.1 Motivation

Intelligent transportation systems (ITSs) are a significant element of smart cities. In recent years, autonomous vehicles (AVs) have attracted much attention due to their potential to revolutionize mobility and safety in transportation. AVs will share urban roads with human traffic participants in smart cities [131]. Thus, to drive safely and efficiently without human intervention in a complex environment, AVs must evaluate their driving intelligence while interacting with their surroundings and correctly estimate the behavioral intention of other participants (i.e., AVs or human drivers).

The main research methods on intention estimation and behavior prediction are composed of machine learning (ML) based on classical probability (model-driven) and deep learning (data-driven) [88, 148]. Traditional ML algorithms generally respect the evolution process of agents' interactive behavior as the property of the Markov decision process (MDP) (i.e., hidden Markov Model) to infer intent [85]. A drawback of traditional ML methods is that due to the high non-linearity of participated agents' behavior intention and their interaction, achieving satisfactory accuracy in performance estimation and predicting the agent behavior is challenging. Moreover, deep learning methods based on data-driven models, such as long short-term memory network, needs a large amount of data to support. If the data quantity is insufficient, it will cause over-fitting, and this method cannot explain the causal relationship between the change of data and the automated driving scene [97]. The real-world scenario is challenging to complete by using the method of artificial statistics received. Furthermore, autonomous driving systems (ADSs) require strong logic and interpret-ability [46]. Therefore, exploring a new computing framework for autonomous driving vehicles is vital.

To achieve a full ADS to accomplish a mission efficiently and, more importantly, shape an adaptive model to a dynamic environment, the AV must be able to learn continuously, modify

its beliefs and make predictions. The cognitive process of brain learning is supposed to be well-referred to the AV's surroundings and situation understanding. The intelligent agent (IA) can perceive its external world and itself using a set of sensors. Accordingly, multi-sensorial information can be divided into exteroceptive and proprioceptive data. Proprioceptive sensors measure the internal agent's parameters, whereas exteroceptive sensors observe the agent's environment. It helps the AV possess the abilities of self-learning and self-referring in different environments to build generative models (GMs) which can understand and explain the AV situation and the external world. Hence, AV can expand its knowledge during continuous learning and evolution and capture novel knowledge to adapt to driving scenarios of different characteristics. Therefore, the multi-sensorial framework equips the AV to incrementally learn self-awareness (SA) models from the perception of itself and the surrounding environment [119].

SA is a broad derived concept from cognitive science (CS) and psychology that describes the property of a system, which has knowledge of its situation using its senses and internal models. Internal and external perception give different forms to the gained knowledge, which is essential for anticipating and adapting to unseen situations. Computational SA methods comprise a promising field that enables an IA to detect non-stationary conditions, learn internal models of its environment, and autonomously adapt its performance and the learning process to the contextual tasks [44].

To this end, we must consider three criteria to achieve high autonomy for vehicles. First, ADS must take human error out of driving actions and respond safely to changing scenarios to improve the safety and efficiency of driving experiences. Second, ADS must be able to understand pre-actions and recognize the driving intentions contained in the behavior. Third, an ADS must be capable of abstracting situational information in a dynamic environment from multi-sensorial information. Therefore, the realization of a full ADS requires solving the following two scientific problems:

- How to make a multi-modal self-awareness model for AVs to understand and assess situations while interacting with the external world, such that AVs can possess the mechanism of memorizing, reasoning, and experiential updating as an expert driver to cope with dynamic changes.

- How to develop an evolutionary and incremental learning system for ADS, where the learning process is similar to the human brain learning procedure and can extend the generalized knowledge learned to unseen scenes.

The key to solving the discussed issues lies in how to introduce the human cognitive model into the framework of an ADS. It is necessary for AVs to constantly interact with their

surroundings and alter their behavior based on changes in the environment. The realization of this process is inextricably linked to the environmental perception of the vehicles, gaining knowledge about their current situation, and making decisions based on this understanding. A widely studied proposition for understanding how the balance between perception and decision-making is maintained is active inference (AIn) [51, 59]. AIn provides a theoretical and practical explanation of how IAs use perception and action to infer hidden states, minimize prediction error and thus reduce their surprise and uncertainty concerning these hidden states. It stipulates a feedback loop between the agent and its interactions with the environment [26]. Explicitly, the agent's actions shape the environment and shift the agent's perception, that influencing future behavior. AIn employs two complementary free energy (FE) functions to measure the surprise, namely, variational FE (VFE) and expected FE (EFE). VFE calculates the fit between the agent's GM and observable outcomes sampled from a generative process (i.e., the environment). Here, the generative process refers to the structure of the environment that generated the agent's observations, and the GM embodies the agent's expectations about the causes of those observations. Conversely, the EFE evaluates possible action trajectories in terms of their ability to reach preferred outcomes via the maximization of extrinsic and intrinsic values. This equips the agent with a formal way to assess different hypotheses about the types of behavior that can be pursued.

Based on this core concept, the AIn framework emerged, presenting a biologically plausible algorithmic approach to agent-based learning and planning. While the fundamental mathematical formulation of AIn is similar to other agent-based decision processes such as Bayesian brain [48] and reinforcement learning (RL) [138], inherent to AIn's formalization is a unique approach to the exploration versus exploitation trade-off, which is an ongoing topic of research in many areas of ML [13]. This, along with its basis in biological plausibility [49, 35], makes AIn an attractive lens to approach autonomous systems (ASs) research. Over the past few years, AIn has solidified itself as a unique mechanism by which to implement IAs to solve complex tasks in complex dynamic environments [125, 50, 95]. In addition, the framework comes equipped with naturally-derived state and parameter exploration schemes. This is in contrast to other ML methods, which require the explicit addition of such schemes to achieve exploratory behavior. Despite much work being done to analyze and develop the field articulately, many areas remain to explore to advance our understanding of its viability and capacity as an ML paradigm. Particularly, theoretical and investigative analysis is needed to ascertain how it compares to other algorithms, mainly when operating under model uncertainty.

In addition, the AIn algorithm presents a framework with great potential for expansion concerning its integration with other ML techniques and expanding upon the core ideas behind

its use of belief propagation and inference. The framework naturally lends itself to effectively representing scenarios of living entities navigating an environment representative of the real-world features. In this regard, the possibilities are endless, and significant opportunities exist to explore the application of the AIn framework to such real-world models.

The presented dissertation aims to investigate some of these opportunities and flesh out our understanding of the AIn algorithm. In particular, the proposed work analyzes how AIn integrates with other learning techniques and employs potential probabilistic models, such as Bayesian methods, for extending the agent's knowledge in a manner that remains congruent with the core concepts of inference and belief propagation.

1.2 Summary of Contributions

Firstly, the thesis starts with a general overview to put AV in its historical context to understand the interests that led to its advent at the design and utilization phases. It enlightens the required knowledge of the state-of-art techniques, which are fundamental to this work. That includes the dynamic representation of the Probabilistic Graphical Models (PGMs). Moreover, a background of concepts and utilizing different learning approaches is provided. It reviews the learning limitation regarding interacting with other participants in a dynamic environment and highlights how the thesis can overcome the constraints by proposing a logical and coherent framework.

Secondly, this thesis presents the benefits of imitation learning (IL) in learning incrementally. This section proposes an adaptive PGM to enable the IA to interact with its surrounding by imitating a set of expert demonstrations, where imitation is not presented only as the implicit repeating but as a process of inference intention through observational learning. Two system models are described to consider agent-tracking scenarios in two cases of interaction with static and dynamic agents, which copes with the core issues of IL in unseen situations.

Thirdly, the thesis provides an AIn framework for incremental imitative learning in ADS that the agent expands its beliefs through experiencing novelty. The proposed generative hierarchical model focuses on representing and minimizing the distinction between the agent's beliefs and the evidence in the external world. I will discuss the connection between Bayesian inference and the transition model on a more conceptual level. The arguments are based mainly on hypotheses regarding perception and acting, as described by inference and optimal action selection. Moreover, another perspective highlights the potential of FE theory at the multi-level of a generalized Dynamic Bayesian Network (GDBN) for the formalization of an action-oriented model as a central process for adaptive behavior under uncertainty.

Finally, the thesis introduces a SA architecture empowered by AIn to improve ADS. A self-aware AS constantly deals with continuous and potentially overwhelming signals from the agent's sensors and their interaction with the dynamic environment. Therefore, learning about interacting with the surroundings dynamically and considering the experienced errors is essential to build a predictive model capturing the novel world's regularities through exploratory actions.

To summarize, the aim of the presented work in this dissertation is to discuss the following research questions:

1. How do incremental learning models be integrated with generative models in ways that offer computational efficiency in autonomous systems?

    - The dynamic interaction between the participants in a non-stationary environment is encoded in a coupled GDBN (C-GDBN) that can be used to facilitate the inference and decision-making processes. It advances a probabilistic computational account of action, observation, and imitation abilities grounded in the active inference framework.

2. To what extent, and in what manner, do the novelty terms affect agent behavior in a complex dynamic environment?

    - A probabilistic framework is presented to solve the exploration-exploitation dilemma by foreseeing actions that minimize the prediction errors and establish a solid foundation for further research on the representation and learning of concepts in a cognitive environment by an autonomous agent.

3. Can the belief propagation mechanism of Bayesian inference be applied to novelty to create a self-aware incremental learning structure that encourages more adaptive behavior?

    - An online evaluation of joint state predictions is applied to update the agent's belief during the back-projection of detected errors at the multi-level of a C-GDBN. We employ a Bayesian sequential decision-making model to distinguish exploration and exploitation processes, which train the A to generate the preferred performance or explore a new course of actions based on its sensory observations and new information provided by the perception of the surrounding.

1.3 Outline of Thesis

- **Chapter 1** presents the introduction, highlights the motivations of the presented research work, and discusses the main contributions of this dissertation.
- **Chapter 2** provides an overview of autonomous vehicles and the levels of autonomy. Moreover, it highlights the weakness and strengths of learning techniques regarding interacting with dynamic surroundings.
- **Chapter 3** describes an online imitation learning framework for autonomous tracking that employs reinforcement learning.
- **Chapter 4** proposes an active inference framework to learn SA module representation of autonomous driving scenarios.
- **Chapter 5** design an exploratory action-oriented model to develop a SA architecture for dynamic interaction with the world.
- **Chapter 6** concludes the presented research work and discusses some open questions for future work.

Chapter 2

Background

2.1 Intelligent Transportation System

ITS refers to a broad range of technologies applied to transportation [90, 117]. ITS combines high technology and improvements in information systems, communication, sensors, controller and advanced mathematical methods provides innovative services related to various modes of transportation (i.e., on the land, on the air, and in the water) and enables users to have a better knowledge of their driving surrounding while using transportation resources in a safe, informed, and coordinated manner [86, 137]. ITS plays a vital role in developing smart cities that are being developed with higher accuracy. AV technology is projected to decrease road incidents, improve travel costs and congestion, and alleviate climate change.

2.1.1 Autonomous Driving

One of the most significant contributions towards the ITS has been the development of autonomous driving (AD). The definition of an autonomous driver is driving a vehicle from one place to another place without a human controller. Avs perceive the environment through senses using various sensors, and then this information is utilized to drive without the need for any human intervention [105]

Since before the twenty-first century, there has been much enthusiasm for autonomous cars. Researchers and industry leaders have been competing to develop the first fully autonomous vehicle that is robust, reliable, and safe for the real world, including high-speed driving environments. For an AV to navigate effectively, technologies from multiple disciplines need to combine. These disciplines broadly include computer science, electrical engineering, and mechanical engineering [133]. Linriccan Wonder of the 1920s was the first radio-controlled car. In 1939, electric cars powered by embedded circuits were showcased.

The advent of a robotic van happened by Mercedes-Benz in 1980 that used vision-guided systems, which was the starting point for high technologies such as lane keep assist and lane departure warning.

Significant contributors to early AV research can be attributed to AV tests and competitions held worldwide. These competitions provided opportunities for industry and researchers to assess the capabilities and boundaries of AVs in various driving environments. However, more importantly, they identified major difficulties and shortcomings in AVs, some of which still need to be solved.

One of the first long-distance AV road tests, No Hands Across America, was introduced in 1995 [15, 115]. This event pushed the boundaries of AV technology requiring the AV to steer across the United States while the human drivers controlled the vehicle's acceleration and braking. Around the same time, an AV drove from Germany to Denmark in the Munich to Odense UBM Test [15, 92]. In 1998, an AV journeyed through Italy's rolling hills, and unpredictable weather conditions in the ARGO Project [14, 15, 23]. In each of these tests, the AVs drove autonomously for 90–98% of the journey using primitive lane departure warning systems, lane-keeping systems, and inter-distance regulation systems [14].

Moreover, vehicle developers noted through these tests that many areas of AV technology require significant improvement. These areas included mostly perception techniques, driving in complex, urban scenarios, and improving erroneous obstacle and road marking detection [11, 14, 15].

Significant developments were made in the early 2000s when the lane departure warning system, adaptive cruise control, self-parking assistance, auto-pilot, and traffic sign recognition were developed for AVs [145, 149, 78]. In 2003 a competition by the Defense Advanced Research Projects Agency (DARPA) was initiated, which required vehicles to drive without the aid of road markings through an off-road desert course [123]. In the first DARPA Grand Challenge (in 2004), no vehicle could complete the course, but in the following challenge (in 2005), five vehicles completed their mission [27]. After the DARPA Grand Challenges, AV research steadily increased, and researchers began to address the challenges of driving in complex environments. Although the DARPA Challenges tested the AVs in almost complicated scenarios, these challenges still lacked certain aspects of a real-world urban driving scene, such as roadway obstacles.

Since the DARPA challenge, several automated driving competitions and trials have been held. Relevant examples included the VisLab Intercontinental Autonomous Challenge [24] and the Hyundai Autonomous Challenge [30] in 2010; the Intelligent Vehicle Future Challenge [146] from 2009 to 2013; the Proud-Public Road Urban Driverless Car Test, in 2013 [25]; the Grand Cooperative Driving Challenge (GCDC) [45], in 2011 and 2016; the

Autonomous Vehicle Competition, from 2009 to 2018; and the European Land-Robot Trial (ELROB) [129], which since 2006 is held yearly.

Although each of these challenges developed promising advances in AV technology in academia and industry worldwide, industrial leaders acknowledge that AVs are not yet robust enough (fully autonomous) to drive without human supervision. To gauge AVs' autonomy level (described in the following section), the Society of Automotive Engineers International published a classification system that the level of autonomy may range from level zero to five based on the human driver intervention and attentiveness required by them.

2.1.2 Levels of Autonomy

There is an expectation that urban transportation will include a mix of manual, semi-automatic, and full AVs by 2025 [111]. To discuss the possible application of driving automation, the society of automotive engineers (SAE) [72] defined six levels of automation as followings:

- **Level 0** has no autonomy. The human driver performs all driving tasks, even when augmented by warning systems or intervention mechanisms.
- **Level 1** is controlled by the human driver, but the driving assistance system executes some minor driving modes.
- **Level 2** is combined with the automated functions, i.e., acceleration/declaration and steering, using the observed information about the vehicle's surroundings. However, the human driver remains engaged with the dynamic driving tasks.
- **Level 3** is not required the human driver to monitor the environment. However, there is the expectation that the human driver responds appropriately to a request to intervene.
- **Level 4** can perform all driving functions by an ADS under certain conditions, even if the human driver does not respond appropriately to an intervention request. If the human driver fails to take control of the vehicle, the ADS steers the vehicle to the side of the road in a controlled manner to stop it.
- **Level 5** has full autonomy. AV can perform all dynamic driving tasks in any condition. The human driver still has the option to control the vehicle.

Based on the above discussion, only level 4 and level 5, which are high and full automation, involve complete automation of the driving task and exclude human intervention during automation. Outside these areas, the human driver is still required. Only level 5 automation would consist of driverless operation anywhere. See Table.2.1, which outlines key features of partially-automated and fully autonomous vehicles.

Table 2.1 Brief overview of level of automation in vehicles.

Level	Name	Human centered/ Autonomous	Steering and speed	Monitoring of driving environmnet	Fallback performance of driving tasks	Saystem capability
0	No automation	Human driver is in charge of all driving tasks	Human driver	Human driver	Human driver	No system-driven capability
1	Driver assistance	Human driver is in charge of all driving tasks	Human driver and ADS	Human driver	Human driver	Some minor driving modes
2	Partial automation	Human driver is in charge of all driving tasks	ADS	Human driver	Human driver	System driven steering and acceleration/ deceleration
3	Conditional automation	System assists human driver in some non-critical driving tasks	ADS	ADS	Human driver	Human driver expected to respond when need arises
4	High automation	Human driver is in charge of a few of driving tasks	ADS	ADS	Human driver and ADS	Human driver only need to intervene if unavoidable
5	Full automation	AS is in full charge of all driving task	ADS	ADS	ADS	All driving modes

2.2 Self-Awareness in Autonomous Vehicle

A full AS requires sophisticated self-assessment capabilities to be fail-operational in different scenarios. Therefore, making decisions and potentially taking actions without direct human intervention requires some knowledge about the system and its environment. Self-awareness (SA) is a crucial ability for a system to effective management and adapt to changing circumstances.

SA in a computing model refers to a paradigm for a system that collects information to maintain knowledge about its own internal states, possible actions, and the result of these actions on the system and its surrounding. This paradigm is appropriate for advanced intelligent decision-making in a dynamic environment under uncertainty, which facilitates explainable autonomy and self-adaptation [84].

Over the years, SA has been the objective in the fields of psychology and CS [6, 5]. More recently, the concept of SA has been transferred to artificial agents aiming at designing IAs and analyzing their behavior to enrich the capability of autonomy in different fields, including ML and robotics [144, 120]. SA is a promising ability that allows an autonomous agent (i.e., AV) to learn the internal model of its surroundings through non-stationary conditions to adapt its behavior and structure to contextual tasks autonomously.

An autonomous agent perceives the environment using the multi-sensorial information from the exteroceptive and proprioceptive sensors to focus on perceptions of both external and internal models, which is essential to anticipate and adapt to unknown situations. Proprioceptive sensors measure the internal agent's parameters, whereas exteroceptive sensors observe the agent's environment. An autonomous agent equipped with a SA representation can be introspective at various levels of hierarchy using obtained joint and dynamic sensory data analysis. From the agent's perspective, introspection is associated with estimating and representing dynamic causal relationships based on observed sensory information [119]. Such representation enables the agent to model the dynamic interaction between itself (i.e., proprioceptive information) and its surroundings (i.e., exteroceptive information).

Therefore, a SA autonomous system requires initialization, model creation, inference, anomaly detection, and the decision-making policy as the main capabilities [119]. Initialization refers to the early knowledge of an IA about its surroundings. To create the learning model, the agent encodes the gained experiences to facilitate predicting the future states and the posterior comparison with evidence. In a predictive model, inference shows how the agent predicts its own future states and surroundings depending on its current state. In order to detect abnormalities, the agent recognizes new observations that the previously observed situations cannot explain. Consequently, a dynamic decision-making policy must be employed to regulate the agent's actions.

In the case of AVs, SA capabilities have yet to be sufficiently studied. This study aims to capitalize on these core capabilities to achieve reasonable autonomy for operating in dynamic, interactive, and uncertain environments. The following section highlights the state-of-arts methods used in this dissertation to develop an autonomous self-awareness system for AVs by adapting the inferences and learning incrementally from the experiences.

2.3 Dynamic Representations for Autonomous Driving

2.3.1 Generative Model

One of the strong candidates towards developing algorithms that can analyze and understand data are generative models (GMs) [62]. A GM is a probabilistic description of generating predictions about observations [19]. Generative models demonstrate how a set of observed data could have arisen from a set of underlying causes. Such models have been commonly employed in brain learning approaches to make inferences about the causes of various conditions or solve a complex inference problem, where it must select the best hypothesis about the external world based on the sensory data.

Here, we are concerned with GMs explaining how the Bayesian brain works. Bayesian inference relies on the GM to formalize beliefs about how outcomes are caused. GMs are capable of both generating synthesized data and providing a distance metric to measure the deviation of the predicted data from the observed one. These statics models aim to learn the joint distribution over a set of random variables from which data samples can be generated from that distribution. A generative process describes transitions among states in the agent's environment that generate observed outcomes. These states are referred to as hidden because they cannot be observed directly. Their transitions depend on action, which depends on posterior beliefs about the next state. In turn, these beliefs are formed using a GM of how observations are generated. The GM describes the agent's beliefs about its surroundings, where expectations encode beliefs about hidden states and policies [21, 4]

In order to achieve safe and high-quality decision-making and motion planning, autonomous agents should be able to generate accurate probabilistic predictions from an uncertain environment. Probabilistic Graphical Models (PGMs) are a specific class of GMs providing stochastic behavior modeling of interacting variables whose relationships are represented in a graphical structure, which is explained in the followings.

2.3.2 Learning Low-Dimensional Representations

Learning low-dimensional representations reduces the dimensionality of the observation space while maintaining the characteristics of the data. Low-dimensional representations can also help reveal latent structures, allowing for deeper insights into the observations. Therefore, GMs are proposed that allow learning low-dimensional representations of the observations, providing means for analyzing the observed data.

Besides the size reduction of the data to increase the efficiency of the post-processing steps, low-dimensional representations can also be used to learn the structure of the data. In many applications, features have a specific meaning and can, therefore, be interpreted by experts. If an IA is able to learn the features present in the data in compliance with its prior assumptions, it can obtain a deeper understanding of the observations [18, 66].

This thesis proposes an autonomous learning framework to improve post-processing steps and allow for accurate data analysis. The obtained observations via the IA (i.e., AV) is a random variable that can be defined into the state representation to encode into probability distributions. This model the interactions and causal effects between the random variables. The following subsections investigate statistical methods for low-dimensional data acquisition and Bayesian models to identify the latent structure of the data. The latent state space (i.e., extracted features) can be represented as a dynamic probabilistic model by utilizing the well-renowned Bayes' theorem.

2.3.3 Probabilistic Graphical Model

A probabilistic graphical model (PGM) defines a family of represented probability distributions in terms of a directed or indirect graph. The graph's nodes represent random variables, and its structure translates into statistical dependencies among them that drive the computation of joint, conditional, and marginal probabilities. [75]. In applications, most random variables are chosen to express the variability of an observed quantity, such as the expression of a specific state measured under a particular condition. Some random variables, however, may specify unobserved quantities that are believed to influence the observable outcomes of a given experiment, such as which cellular processes were active when measurements were taken. The relationship of the graph specifies the hypotheses about how observable and latent quantities influence one another. Moreover, random variables are explained by constants underlying their distributions. These constants are referred to as frequentist parameters and hyper-parameters in the Bayesian paradigm [1].

Bayesian Network (BN) is a popular class of PGMs that is introduced by [113], where the graph and the probability theories are combined with modeling a comprehensible representation of the joint probability distribution. The BNs point out useful modularities in the underlying problem and help accomplish decision-making tasks, especially in uncertain domains.

2.3.4 Dynamic Bayesian Network

BNs do not model temporal relationships among variables. However, the temporal dependencies of interactions that involve different interactive behaviors at different time scales are consequential in AS. A Dynamic Bayesian Network (DBN) is an extended BN that is able to model influences over time series or sequences [99, 64].

A DBN consists of a graph containing directed links between involved variables, not allowing loop cycles between a given variable. DBN represents the motion of observed agents in an environment and includes dependencies between involved random variables as time evolves. It facilitates the representation of different inference levels related to agents' dynamics and incorporates the variables' uncertainties when predicting future instances. Therefore, DBN allows encoding probabilistic dependencies and feedback between random variables over different time slices, which provides inferences about the system.

One of the main advantages of DBNs is their hierarchical nature; a DBN allows us to express causal relationships between high-level variables (capturing abstract semantic information about the world) and low-level distributions (capturing rough sensory information about the environment). DBNs model hierarchical relationships between different variables

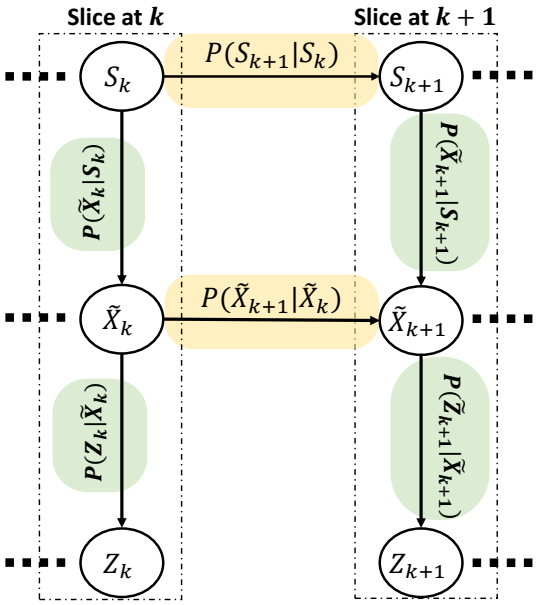


Fig. 2.1 General structure of Dynamic Bayesian Network.

with their respective evolution through time. At each time instant (DBN's slice), causal relationships between variables are encoded through inter-slice links. On the other hand, causal relationships between variables in subsequent time instances are encoded through temporal links.

Fig.2.1 shows a DBN representation that includes three hierarchical variable levels at each time slice. In this work, the lowest level of inference corresponds to the agent's observation measurements (Z_k). The medium inference level $(\tilde{X}_k)$ represents states that capture the agent's continuous information. The top level of the hierarchy (S_k) corresponds to the super-states, which consist of the agent's dynamic that can be considered as a set of discrete sub-tasks performed one after the other. Each sub-task is described as a linear model that defines the agent's expected dynamics according to their state in the environment.

Each time slice of the proposed representation in Fig.2.1 involves three conditional dependencies as follows. $P(Z_k|\tilde{X}_k)$ explains the probability of obtaining an observation given the agent's state. $P(\tilde{X}_{k+1}|\tilde{X}_k)$ represents the probability of obtaining a future state given its current one, and $P(\tilde{X}_k|S_k)$ expresses the probability of having the agent's state $\tilde{X}_k$ given the super-state S_k. The proposed DBN model is a generative model where continuous states are encoded inside discrete regions. DBN makes spectrum inferences at two levels, discrete and continuous, through Bayesian Filtering.

2.3.5 Bayesian Filtering

Bayesian theory is a branch of mathematical probability theory that is able to model the uncertainty about the world by incorporating prior knowledge and observational evidence. In Bayesian inference, states and parameters are identified as random variables, which can be either deterministic or time-varying [32]. The causal and prior knowledge is used to infer conditional probabilities at hierarchical levels based only on observations. Graphical models (i.e., DBNs) allow the construction of hierarchical statistical models and illustrate the Bayesian inference.

Bayesian filtering can be applied based on two assumptions. one is, states (x) are following a first-order Markov process that can be expressed as $P(x_k|x_{k-1})$, and the other one is that observations are independent of the state. The filtering process involves extracting information about a quantity of interest at time k and using observed data up to and including that time. Accordingly, the purpose of Bayesian filtering is to compute the marginal posterior distribution of the state x_k at each time instant k given the observation measurements up to the time step k as $P(x_k|z_k)$ [126]. The fundamental steps of Bayesian filtering are given in the following steps.

- Initialization. The filtering procedure starts from the prior distribution $P(x_k)$.
- Prediction. The predictive posterior of x_k, given the dynamic model. can be computed by the Chapman-Kolmogorov equation as:

$$P(x_k|z_{k-1}) = \int P(x_k|x_{k-1})P(x_{k-1}|z_{k-1})dx_{k-1}, \tag{2.1}$$

- Update. Given the observation measurement at each time instant (z_k) the posterior of state x_k can be updated using the Bayes' rule as:

$$P(x_k|z_k) = \frac{1}{Z_k}P(z_k|x_k)p(x_k|z_{k-1}), \tag{2.2}$$

where Z_k is the normalization constant given as:

$$Z_k = \int P(z_k|x_k)P(x_k|z_{k-1})dx_k. \tag{2.3}$$

The Kalman Filter (KF) [77] is a specific case of the Bayes filter employed where there is uncertain information about a dynamic system. KF is a closed-form solution for the Bayesian filtering model that comprises the prediction and update steps. A KF can be modeled with state transition probability $P(x_k|x_{k-1})$ and a measurement probability $P(z_k|x_k)$ using physical

laws of motion to make inferences of the future beliefs where the dynamic system and measurements are linear Gaussian. This assumption holds if the model's noise follows the multivariate normal distribution. On the other hand, Sequential Monte Carlo methods, in particular the Particle Filter (PF), reproduce the work of the KF to compute the estimates based on random and weighted samples (particles) to deal with non-linear and non-Gaussian environments [108].

A combination of KF and PF can provide a probabilistic switching graphical model to infer posterior probabilities on discrete and continuous states iteratively. The combined approach is called Markov Jump Particle Filter (MJPF) [10]. MJPF consists of a PF working at the discrete level, embedding in each particle a KF at the continuous state level. Consequently, each particle has an attached KF, which depends on the superstate S_K. Therefore, MJPF can be linear/non-linear Gaussian/non-Gaussian.

2.4 Learn Action-Oriented Models

Converging neuroscience and ML approaches suggest decision-making models that efficiently adapt to the environment by exploiting the probabilistic model of their world [41, 22, 37]. These models encode statistical representations of the states and contingencies in an environment and agent-environment interactions where decisions are based on prospective evaluation of potential action outcomes. For instance, probabilistic models can perform perceptual inference, implement anticipatory control, overcome sensory noise, and generalize prior knowledge to new circumstances. Despite the advantages of encoding a probabilistic model, dynamic environments are extremely complex and infeasible to model them. Therefore, it is essential to equip the decision-making models with action control to enable adaptive behavior rather than accurately presenting the surrounding. We refer to such models as action-oriented, which do not need to model their environment [7] exhaustively. By reducing the need for models to be isomorphic with their environment, an action-oriented approach can increase the tractability of the model learning process [140, 124, 139, 89]. Learning from experiences still is a challenge within an action-oriented approach. In the followings, we explore the learning approaches for adaptive behavior in dynamic environments.

2.4.1 Reinforcement Learning

Reinforcement learning (RL) [138] is one of the ML techniques that study learning from experience and improve its performance at a specified task. The agent experiences the environment's states by performing an action. The action changes the environment's state.

Also, the state transitions are evaluated via a reinforcement signal defined as a reward. The RL environment is generally treated as MDP [12], which can be represented as $(\mathrm{S}, \mathrm{A}, P, \mathrm{R}, \gamma)$, where S denotes the state space, A denotes the action space. $P(s'|s,a)$ represents the transition probability from s to s' given action a, where $s, s \in \mathrm{S}$ and $a \in \mathrm{A}$. R is the reward function, and $\gamma \in [0,1]$ is the discount factor. An abstraction of the reward signal is the value function which represents the benefit of being in a specific state. It can learn the optimal behavior over time by systematic trial and error. The RL algorithm aims to discover the action policy that maximizes the long-term reward value by interacting with the environment.

Typically, the learning agent is not explicitly supervised on how to act by an optimal expert, but it must learn the best action that brings the highest reward value in the trial and error procedure, the so-called exploration phase. As the learning agent gathers knowledge to distinguish a good performance from a wrong one, it can exploit the gained knowledge to make a decision. The challenge of RL is to design the best trade-off between exploration and exploitation, that is, the capability of an agent to use its knowledge to obtain high rewards and, by contrast, being capable of exploring new possibilities in such a way as not to remain stuck in a local optimum.

2.4.2 Imitation Learning

In recent years, the demand for IA capable of mimicking human behavior has grown substantially. In recent years, advances in AS have led to a wide range of potential applications that require an agent that can make intelligent decisions and perform realistic motor actions in various situations. Many future directions in AVs rely on the ability of the agent to behave as an expert would when presented with the same situation. Previous researches show that employing prior knowledge provided by an expert is more effective and efficient than learning from scratch [3, 128, 17]. One of the intuitive ways of transferring expert knowledge is to provide optimal demonstrations for the desired behavior that the learning agent (L) is required to accomplish [118]. Imitation learning (IL) refers to the occurrence of skills or behaviors by observing an expert demonstrating a given task. IL is essential to machine intelligence with inspiration and basis stemming from neuroscience. It has from an early point been viewed as a critical part of the future of IA [128].

Similar to standard supervised learning, where examples represent pairs of features and labels, in IL, the examples demonstrate pairs of states and actions where the state represents the agent's current situation and the status of a target object if one exists. A typical IL workflow starts by acquiring sample demonstrations from an expert agent (E), which are then encoded as state-action pairs. These examples then train a policy (π^E) that follows a certain distribution. A dataset $(\mathscr{D})$ is acquired from the (E)'s actions and implicitly its policy

to train the learner policy (π^L) as:

$$\pi^E \rightarrow \mathscr{D} \rightarrow \pi^L, \tag{2.4}$$

IL aims at minimizing the distinction between the expert agent and the learning agent state distribution in order to build the learner's policy as follows:

$$\pi^L = \underset{q,p}{\operatorname{argmin}} \Delta(q(s), p(s)), \tag{2.5}$$

where $q(s)$ is the distribution of the states induced by the experts' policy and $p(s)$ is the distribution of states induced by the learner, and $\Delta(q, p)$ measure the distinction between q and p.

However, more than learning a direct mapping between state and action is needed to achieve the required behavior. This can happen due to several issues, such as errors in acquiring the demonstrations or insufficient demonstrations [71]. Moreover, the task performed by the learner may vary slightly from the demonstrated task due to changes in the environment, obstacles, or targets. Therefore, IL frequently involves another step that requires the learner to perform the learned action and modify the learned policy according to its task performance. This self-improvement can be achieved concerning a quantifiable reward or learned from examples. Many of these approaches fall under the broad umbrella of RL.

Moreover, a learner could very well arrive at a suitable solution that achieves a particular quantifiable goal but differs significantly from how an expert would approach the task. This is necessary for many AS domains that the learner's performance is only as good as an expert observer's perception of it. Therefore, teaching a learner the desired behavior from a set of collected instances is favorable. However, more than the imitation of the expert's motion is often required due to variations in the task, such as the positions of objects or inadequate demonstrations. Therefore, IL techniques need to learn a policy from the demonstrations that can generalize to unseen scenarios. As such, the agent learns to perform the task rather than deterministically repeating the expert's behavior.

2.4.3 Active Inference

In the recent past, a paradigm shift in computational and CS toward the Bayesian brain approach is considered, which conceptualizes the brain as a prediction and inference machine, actively trying to predict, experiment with, and understand its surroundings [112, 132]. Active inference (AIn) is a unified mathematical framework for modeling perception, learning, and

decision making [36, 60]. AIn considers the interactions between these processes as an interdependent inference. IA infers the probabilities of external states and the environmental consequences by involving the prior beliefs with the sensorial observation. The inferences underlying decision-making is active because the agent infers the actions most likely to generate preferred sensory input. IAs also infer the actions most likely to reduce uncertainty and facilitate learning. This leads the decision-making procedure to the expected actions that optimize a trade-off between maximizing reward and information gain. Hence, AIn predicts match behavior and perception with empirical observations [134, 135]. Therefore, AIn consists of three crucial components:

- A generative model of the agent's environment,

  Optimal predictions are based on sensorial evidence that is evaluated concerning a GM of the observed outcomes. Hence, the behavior can be framed in terms of presenting the predictions prescribed by prior preferences [55].

- Fit the model to sampled observations to reduce surprise,

  The GM contains beliefs about future states and policies, where the most likely policies lead to preferred outcomes. The AIn framework assumes that perception and learning can be understood through computing a quantity known as VFE [68] to minimize the divergence between the prior and posterior. The previous beliefs refinement considers an explicit representation of past and future states conditioned on the learning policies that lead to updating the Bayesian beliefs and the context of learning. Therefore, beliefs inform the agent about the future (i.e., prediction) and the past (i.e., prior knowledge).

- Select an action that minimizes uncertainty,

  Action selection, planning, and decision-making can be understood as minimizing EFE, which quantifies the VFE of various actions based on expected future outcomes. Minimizing FE concerning expectations of hidden states and parameters ensures that they encode posterior beliefs, given observed outcomes [54]. This enables action to realize preferred outcomes based on the assumption that both action and perception try to maximize the GM's evidence or likelihood, as FE scored.

2.4.4 Incremental Learning

One of the significant challenges of learning approaches in a dynamic world is learning new situations and expanding the agent's knowledge incrementally, where new situations are observed over time. Therefore, algorithms that can process and understand new concepts

from such data are required. This leads to the concept of incremental learning, allowing the agent to acquire new knowledge while preserving the previous ones [147, 31].

Incremental learning is an essential capability for brain-like intelligence as biological systems are able to learn through their lifetimes and accumulate knowledge over time continuously. It is an ML paradigm where the objectives of ML research are transforming prior knowledge to the currently received data to facilitate learning from new observations, accumulating experience over time to support decision-making, and achieving global generalization through learning to accomplish tasks. During the incremental learning situation, raw data from the environment with which the IA interacts become incrementally available over an indefinitely long learning lifetime. Therefore, the learning process fundamentally differs from the traditional static learning process, where representative data distribution is available during training to develop the decision boundaries. IA should be equipped to automatically modify its knowledge to learn new data distributions. Therefore the learning approach should meet the following criteria as an incremental learning approach.

- The IA is able to learn, modify and update its beliefs,
- The learning model preserve previously acquired knowledge,
- The model is generative; it is able to generate new data or merge the existing ones as needed,
- The GM is dynamic; it adapts to the changing environment.

2.4.5 General Comparison

AD requires the resolution of perception and motion planning issues in the presence of dynamic objects interacting with the environment. Some learning approaches are introduced and discussed in the previous sections, which are compared in the following.

The complex interactions between multiple agents are significant challenges due to the difficulty of predicting their future motions. Most model-based AD approaches necessitate manually designing the driving policy model [109, 65] or they are equipped with safety assessments to assist the human driver [110, 142]. While designing a decision and planning system for AVs is complex, an alternative is to learn the driving policy from an expert agent using IL. The existing works of the IL approach for driving can handle simple driving tasks such as lane following [107, 33]. However, if the agent is dealing with a new environment or a more complicated task (such as line-changing), it is required that the human driver has to take control, or the system fails ultimately [20, 127]. More particular, a typical IL procedure is direct learning, where the main goal is to learn a mapping from states to actions that

Table 2.2 Comparison with existing methods from literature.

functionalities	[20]	[33]	[43]	[61]	[82]	[107]	[110]	[127]	[130]	[141]	[142]
Indirect learning	✗	✓	✗	✓	✓	✓	✓	✗	✗	✗	✓
No expert intervention	✗	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗
Self-improvement	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
Adapt to dynamic environment/behavior	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
Incremental learning	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	✓
Calculate future EFE/reward	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

mimic the demonstrator explicitly [141, 43]. Direct learning methods are categorized into classification methods when the learner's actions can be classified into discrete classes [87, 2], and regression methods which are used to learn actions in a continuous space [122]. Direct learning often is not adequate to reproduce proper behavior due to issues such as insufficient demonstrations and performing a different task due to environmental changes. Besides, indirect learning can complement direct approaches by refining the policies based on sub-optimal expert demonstrations [61].

The critical drawbacks of IL are that the policy never exceeds the suboptimal expert performance and that the learning policy is vulnerable to distributional shift [104]. Therefore, IL frequently involves another step that requires the learning agent refinement of the estimated policy based on its current situation. This self-improvement can be achieved by a quantifiable reward or learned from instances. Many of these approaches come under the RL methods. RL allows encoding desired behavior — such as reaching the target and avoiding collisions — and relies not only on perfect expert demonstrations. In addition, RL maximizes the overall expected return on an entire trajectory, while IL treats every observation independently [82], which conceptually makes RL superior to IL. RL does not have prior knowledge from an expert. Therefore the learning agent has no clue to realize desired behaviors in sparse-reward settings [130]. Even when RL succeeds in reward maximization, the policy does not necessarily achieve behaviors that the reward designer has expected. In addition, learning through trial and error requires reward functions designed specifically for each task. Defining rewards for such problems is complex and still unknown in many cases.

Behavior learning, such as IL and RL, would be complex without representation or model learning from the environment. To overcome the mentioned limitations, AS employs a GM of the world and computes the mathematical amount of FE to explain perception, action, and model learning in a Bayesian probabilistic way [53, 58], that can handle behavior learning and model learning at the same time in a dynamic environment. Table. 2.2 demonstrates a comparison of the main functionalities of autonomous driving.

Chapter 3

Incremental Learning using Imitation Learning

3.1 Introduction

IL has been recognized as a promising technique to teach autonomous agents advanced skills. It is based on the idea that IA learns new behaviors by observing and imitating other agents' movements. Observation, representation, and reproduction are three crucial steps in the imitation procedures. The standard programming approaches by expert demonstrations have usually been focused on reproducing the demonstrated behavior [17]. We aim to enable the IA to interact and imitate, where imitation is not presented only as the implicit repeating but as a process of inference intention where the agent learns from its observations. This imitative behavior is a significant prerequisite for having IA capable of advanced interaction with the world and adapting and learning from it. Considering SA and CS hypotheses, corroborated by several empirical findings, postulate that the understanding of the external world is achieved by employing the agent's internal model [121].

The imitator agent must be able to describe an action's intention from its observation. Recognizing the intention of an acting agent could be computationally interpreted as a matching problem between the expectation and the observation. [39] describes descriptive and generative approaches for classifying an observation. In the former, low-level features are extracted from the learning agent's observation and then differentiated from the pre-existing knowledge to generate the corresponding action to the current representation. In the latter, generative approaches using a set of latent variables encode the causes capable of producing the observed data. Billard et al., in [17], define two typologies of skill representation: trajectory-level encoding, which is represented as a non-linear mapping between sensory and

motor data, and symbolic-level encoding where tasks are described symbolically using ML methods. ML techniques facilitate representing hierarchies of behaviors and sequences of states.

In this section, we propose an adaptive PGM which copes with the core issues of IL (i.e., observation, representation, and reproduction of skills). The presented model is based on learning a Generalized Dynamic Bayesian Network (GDBN), which is able to characterize structured behaviors at different levels of abstraction hierarchically, and also grows by learning new skills (i.e., new observations) or modifying the prior representation incrementally.

3.2 Model I - Single Dynamic Agent

Motivated by the previous discussion, we propose a framework for autonomous tracking in a continuous environment that combines IL with RL. IL is a pre-training step to encode an expert demonstration in a GDBN that describes desired behaviors. This work employs the discrete information of a probabilistic expert model enabling the learning agent to improve its actions by minimizing the imitation cost that allows for avoiding abnormal states in the future. The proposed approach includes two main phases: offline and online learning. In the former, we learn a reference model encoding the dynamical behaviors of an expert agent (E) moving to a fixed goal G. In the latter, an incremental IL model is learned where a learning agent (L) attempts to learn sub-optimal behavior by observing E demonstrations and updates its knowledge while transiting in a continuous environment to reach G.

3.2.1 Learn a Generalized Dynamic Bayesian Network

The offline learning process aims to learn a GDBN based on E behavior which can be used as a reference model by L. The proposed GDBN represents the E's dynamics in the environment and models hierarchical relationships between different variables with their respective evolution through time graphically.

The GDBN model consists of three levels:

- Bottom level. It depicts the E's observations represented by Z_k^{E}.
- Middle level. It encodes the continuous information where the generalized states (GSs) are expressed as:

$$\tilde{X} = \{\tilde{X}_k\}_{k=1,\dots,k}, \tag{3.1}$$

where $\tilde{X}_k = [X_k,\ \dot{X}_k]^\intercal$, $\dot{X}_k \sim \frac{X_k - X_{k-1}}{\Delta k}$ and Δk is the sampling time.

- Top level. It represents the E's discrete states $(\mathrm{S}_k^{\mathrm{E}})$ that explains the dynamical transition behaviors reflected into a discrete semantic space.

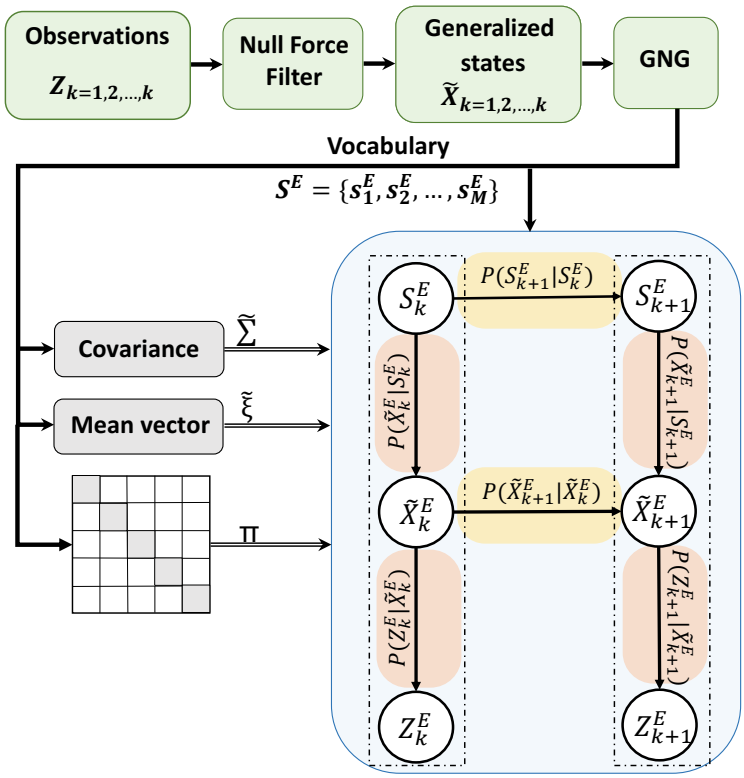


Fig. 3.1 Learning DBN using expert demonstrations. Inter-slice links are depicted in orange, and temporal links are colored in yellow.

As the graphical representation depicted in Fig. 3.1 shows, Vertical arrows describe causalities between continuous and discrete inference and observed measurements. Horizontal arrows represent temporal causalities between hidden variables.

We assume the observed signalZ_k is a linear combination of the latent GS $\tilde{X}_k$ that represents the direct cause of the observation and a multivariate Gaussian noise. The observation model that maps $\tilde{\mathrm{X}}_k^K$ to Z_k^K is defined as:

$$\mathrm{Z}_k^E = \mathrm{H}\tilde{\mathrm{X}}_k^E + v_k, \tag{3.2}$$

where $H(.)$ is the matrix that maps hidden generalized state to the observed data and v_k is the measurement noise which is assumed to be zero mean Gaussian noise with covariance R_k such that $v_k \sim \mathcal{N}(0, \mathrm{R})$.

We assume that the dynamics of GSs evolve according to a static equilibrium assumption described as:

$$\tilde{\mathrm{X}}_k^E = \mathrm{A}\tilde{\mathrm{X}}_{k-1}^E + w_k, \tag{3.3}$$

where $\mathrm{A} \in \mathbb{R}^{d\times d}$ is the dynamic matrix. $\mathrm{A}\tilde{\mathrm{X}}_k$ takes the state-space information from $\tilde{\mathrm{X}}_k$ and makes null its time derivatives. The variable w_k is a zero-mean Gaussian distribution representing the process noise f the dynamical modeling. This implies a null acceleration, and the corresponding dynamic model follows static equilibrium called Null Force Filter (NFF) [74]. An NFF can be interpreted as an unmotivated KF. NFF uses the innovations obtained by observing a sequence Z_k^E to estimate the next state that describes the agent's motion in the GS space. The innovations can be seen as generalized errors (GEs) which are the mismatches between observations and predictions as:

$$\tilde{\varepsilon}_k = \mathrm{H}^{-1}\big(\mathrm{Z}_k^E - \mathrm{H}\tilde{\mathrm{X}}_k^E\big). \tag{3.4}$$

The GEs that capture the real dynamics of the signal are clustered in an unsupervised manner using the Growing Neural Gas with utility measurement (GNG-U) [73]. GNG-U outputs a set S^E of discrete variables (i.e., clusters) representing the discrete level of the GDBN structure and forming the so-called Vocabulary such that:

$$S^E = \{s_1, s_2, \ldots, s_M\}, \tag{3.5}$$

Each discrete variable $\mathrm{s}_m \in \mathrm{S}^E$ is assumed to follow a multivariate Gaussian distribution, such that $\mathrm{s}_m \sim \mathscr{N}(\tilde{\xi}_m,\, \tilde{\Sigma}_m)$, where $\tilde{\xi}_m = [\xi_m,\, \dot{\xi}_m]^\intercal$ is the GS centroid of the m-th cluster and Σ_m is its covariance matrix. The discrete level of the proposed GDBN represents the activated cluster ($s_k \in S^E$) at each time instant k. This work assumes that L employs the discrete information in S^E as flash-back memory ($\boldsymbol{D}$) that guides the RL procedure during the online learning phase. Moreover, The probabilistic law that regulates transition among different local forces captured by different clusters can be estimated in different ways (e.g., frequentist or geometrical) and encoded in a Transition Matrix (Π) that estimates the transition probabilities $\mathrm{P}(\mathrm{s}_k|\mathrm{s}_{k-1})$.

$$\Pi = \begin{bmatrix} \mathrm{P}(\mathrm{s}_1|\mathrm{s}_1), & \mathrm{P}(\mathrm{s}_1|\mathrm{s}_2), & \ldots, & \mathrm{P}(\mathrm{s}_1|\mathrm{s}_\mathrm{M}) \\ \mathrm{P}(\mathrm{s}_2|\mathrm{s}_1), & \mathrm{P}(\mathrm{s}_2|\mathrm{s}_2), & \ldots, & \mathrm{P}(\mathrm{s}_2|\mathrm{s}_\mathrm{M}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{P}(\mathrm{s}_\mathrm{M}|\mathrm{s}_1), & \mathrm{P}(\mathrm{s}_\mathrm{M}|\mathrm{s}_2), & \ldots, & \mathrm{P}(\mathrm{s}_\mathrm{M}|\mathrm{s}_\mathrm{M}) \end{bmatrix} \tag{3.6}$$

3.2.2 Imitation Policy

In the online learning phase, we propose an incremental IL model that allows L to learn how to improve the decision-making procedure to reach G by minimizing imitation loss. The learned reference model from E can be employed to predict the expectation in each state and also provide the learning policies to teach the best set of actions ($\mathscr{A}$) that L requires to accomplish its task. Therefore, $\boldsymbol{D}$ leads the active states during new experiences that describe how the agent can act in the environment to change sensory signals in order to match internal predictions of learned GDBN and thus to imitate efficiently by decreasing the abnormalities.

This work suggests training L through the inclusion of $\boldsymbol{D}$ representing E's behaviors, which postulate that L optimizes its movements based on $\boldsymbol{D}$'s predictions over time. RL can formalize the underlying decision-making as an MDP, a model of a discrete-time process wherein an agent's actions may stochastically influence its environment. The proposed approach endows L with the capability of estimating the imitation cost, whereby minimizing the imitation cost (i.e., maximizing rewards in RL) ensures an equilibrium between L and its surrounding. Accordingly, let define L's state and action at a given time instant k as s_k^L, a_k^L, respectively.

We hypothesize that L uses a probabilistic discrete representation S^E that encodes relevant information about the observed E's behaviors (i.e., $\boldsymbol{D}$) instead of exploiting explicitly from E, which rejects the idea of a buffer that replays previously observed E's states $\tilde{X}^E$. The main focus of the online learning phase lies on modeling a dynamic multiple reward function by considering the divergence between L and $\boldsymbol{D}$ at each time instant k to regulate the L's actions in the incremental learning stage.

Two policies are considered, the action-based and the state-based, to evaluate L's performance using the activated cluster ($\mathring{\text{s}}$) from $\boldsymbol{D}$, where $\mathring{\text{s}}$ is the closest E's cluster to the current L's state (s_k^L) calculated by Euclidean distance.

Action-based policy (ρ_a), is employed to minimize the divergence between the performed action by L and the mean action of the activated cluster $\mathring{s}$ from the expert demonstrations, such that:

$$\rho_{a,k} = d_{\mathscr{M}}\big(g_a(\mathring{s}_{k-1}),\, a_{k-1}^L\big), \tag{3.7}$$

where $d_{\mathscr{M}}(\mathtt{X}, \mathtt{x})$ is the Mahalanobis distance [93] between a distribution $\mathtt{X}$ and a point $\mathtt{x}$, $g_a(\cdot)$ is a function that extracts the action-distribution from a GS-distribution, such that $g_a(\mathring{s}_{k-1}) \sim \mathcal{N}(\tilde{\xi}_{k-1},\, \Sigma_{k-1}^{\tilde{a}})$ and Σ_{k-1}^a is the action's covariance information. $\mathring{s}_{k-1} \sim \mathcal{N}(\tilde{\xi}_{k-1},\, \Sigma_{k-1}^{\tilde{a}})$, which can be obtained according to:

$$\mathring{s}_{k-1} = \arg\min_{s_m} \|s_{k-1} - \tilde{\xi}_m\|_2. \tag{3.8}$$

State-based policy (ρ_s), is employed to minimize the distinction between the current L's state s_k^L, which is reached by the last performed action a_{k-1}^L, and the predicted state from the activated cluster after performing the expected action ($\mathring{s}_{k|k-1}$). The discrete probability $P(s_k|s_{k-1})$ from $\boldsymbol{D}$ is employed to estimate $\mathring{s}_{k|k-1}$. The term $\mathring{s}_{k-1}$, required in $P(\mathring{s}_k|\mathring{s}_{k-1})$, is calculated based on (3.8). The state-based policy can be written as:

$$\rho_{s,k} = d_{\mathscr{M}}\big(g_s(\mathring{s}_{k|k-1}),\, s_k^L\big), \tag{3.9}$$

where $g_s(\cdot)$ is a function that extracts the state-distribution from a GS-distribution, such that $g_s(\mathring{s}) \sim \mathscr{N}(\tilde{\xi}_k,\, \tilde{\Sigma}_k^s)$. Σ_k^s is the state's covariance information. The policies indicate the imitation loss regarding a^L and s^L at each time instant k in a continuous range $[0,1]$ that describes the loss value. Global imitation loss $(\mathscr{G}_k)$ takes into account the mean value of both policies as:

$$\mathscr{G}_k = \mathbb{E}(\rho_{a,k}, \rho_{s,k}). \tag{3.10}$$

Hence, minimizing the global imitation loss cause maximizing the learning rate and gaining a high reward.

3.2.3 Action Selection and Update the Model

The learning model uses ε-greedy $\in [0,1]$ as a control input to shorten exploratory behavior over the training episodes. In terms of a high amount of ε, the learning agent motivates to select a random action to explore new states that can be exploited in the future. In the exploiting case, the agent selects the associated action to the maximum value. Thus, the action selection process depends on the ε value whether to explore or exploit, and it is defined as:

$$a_k \sim \begin{cases} \underset{a_k}{\arg\max}\ \mathrm{Q}(\mathscr{A}, \mathrm{s}_k), \text{ if } \varepsilon < \theta \ \text{ (exploitation)}, \\ \textit{random} \ \text{ from } \mathscr{A}, \text{ if } \varepsilon \geq \theta \text{ (exploration)}, \end{cases} \tag{3.11}$$

where $\mathscr{A} = \{a_1, a_2, \ldots, a_8\}$ is a set of eight cardinal and ordinal directions[1] and θ is a defined threshold.

L records the experienced states (s_k^+) along with the performed actions ($a_k \in \mathscr{A}$) in a incremental function $Q(s,a)$ and the new states are saved in set S_Q that grows incrementally

[1]The 8 directions are North, South, East, West, North-West, North-East, South-East, South-West.

as experiences are observed over time:

$$\mathrm{Q} = \begin{bmatrix} \mathrm{P}(a_1^{\mathrm{L}}|\mathrm{s}_1^{\mathrm{L}}) \dots \mathrm{P}(a_1^{\mathrm{L}}|\mathrm{s}^{+}) & \dots \\ \mathrm{P}(a_2^{\mathrm{L}}|\mathrm{s}_1^{\mathrm{L}}) \dots \mathrm{P}(a_2^{\mathrm{L}}|\mathrm{s}^{+}) & \dots \\ \vdots \quad \ddots \quad \vdots & \vdots \\ \mathrm{P}(a_{\mathrm{N}}^{\mathrm{L}}|\mathrm{s}_1^{\mathrm{L}}) \dots \mathrm{P}(a_{\mathrm{N}}^{\mathrm{L}}|\mathrm{s}^{+}) & \dots \end{bmatrix}, \tag{3.12}$$

where $\sum_{n=1}^{\mathrm{N}=8} \mathrm{P}(a_n^{\mathrm{L}}|\mathrm{s}_m^{+}) = 1$ such that s_m^{+} are the new explored states. In order to weigh up the trained model than $\boldsymbol{D}$, L clusters all the recorded pairs $[\mathrm{s}_k^{+}, a_k]$ by employing GNG. The latter outputs a set of clusters representing the new states ($\hat{\mathrm{S}}$) and the corresponding mean actions ($\dot{\mathrm{a}}$), which are added to the updated Q-table (Q^{*}) defined as:

$$\mathrm{Q}^{*} = \begin{bmatrix} \mathrm{P}(\dot{\mathrm{a}}_1|\hat{\mathrm{S}}_1) & (\dot{\mathrm{a}}_1|\hat{\mathrm{S}}_2) \dots \mathrm{P}(\dot{\mathrm{a}}_1|\hat{\mathrm{S}}_M) \\ \mathrm{P}(\dot{\mathrm{a}}_2|\hat{\mathrm{S}}_1) & (\dot{\mathrm{a}}_2|\hat{\mathrm{S}}_2) \dots \mathrm{P}(\dot{\mathrm{a}}_2|\hat{\mathrm{S}}_M) \\ \vdots & \vdots \quad \ddots \quad \vdots \\ \mathrm{P}(\dot{\mathrm{a}}_N|\hat{\mathrm{S}}_1) & (\dot{\mathrm{a}}_N|\hat{\mathrm{S}}_2) \dots \mathrm{P}(\dot{\mathrm{a}}_N|\hat{\mathrm{S}}_M) \end{bmatrix}. \tag{3.13}$$

L adapts the action selection procedure by updating the Q-table defined in (3.12) based on the imitation cost policies at each time instant k. Since the provided Q is a probabilistic table, updating the Q value can be written in a probabilistic form as follows:

$$\mathrm{Q} = (1 - \eta)\mathrm{P}(a_{k-1}|\mathrm{s}_{k-1}) + \eta\Big[(1 - \mathscr{G}_k) + \gamma \max_{a_k} \mathrm{P}(a_k|s_k)\Big], \tag{3.14}$$

where η is the learning rate that controls how quickly the learning agent adopts to the explorations imposed by the environment, $(1-\mathscr{G}_k)$ is the normalized reward measurement with a range in $[0,1]$, and γ is a discount factor.

3.3 Simulation and Performance Evaluation - Model I

3.3.1 Experimental Setup

The proposed framework is validated using a simulated dataset consisting of sensorial information collected by E where it attempts to reach G from different starting points. E moves based on the velocity field model proposed in [29], such that:

$$\vec{G}(r) = \big(\beta - \lambda e^{\frac{-r^2}{\psi}}\big)\hat{r}, \tag{3.15}$$

where r is the distance to G, $\lambda \leq \beta$ and $\hat{r}$ is a unit vector pointing at G. The E positional information and the corresponding velocities are obtained from the odometry module. Sensory data representing positional information from these experiments are used to learn the expert trajectories encoded in $\boldsymbol{D}$ that L uses to imitate E.

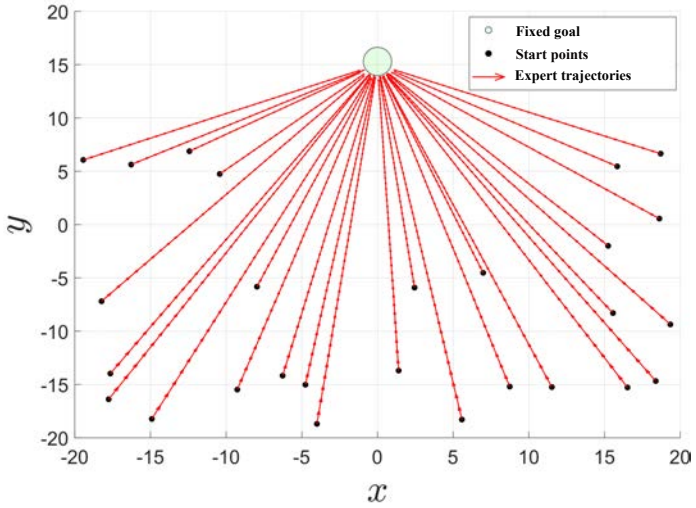


Fig. 3.2 Examples of teacher's actions.

3.3.2 Offline Learning Phase

This section shows the process of learning the flash-back memory ($\boldsymbol{D}$) from E's behavior. Initially, the learned DBN from $\boldsymbol{D}$ is modeled based on the observed expert's actions (see Fig. 3.2). The NFF is used as an initial filter employed on the collocated data during tracking G. NFF outputs the GEs defined in (3.4), which can be clustered using GNG that outputs a set of discrete clusters representing the discrete regions of the trajectories generated by E. Fig. 3.3-(a)-(b)-(c)-(d) illustrate the clusters and the corresponding transition matrix, which L looks at them as a sub-optimal reference information.

3.3.3 Online Learning Phase

During the online learning phase, L modifies its actions based on the learned clusters during the offline phase. Q-table records the L's observations and the corresponding actions as defined in (3.12). The experiments are done in a simulated environment. For having a fair comparative evaluation, all the experiments are considered with fixed steps. We run each algorithm over $4k$ episodes by different start positions to learn how to track G through learning the imitation policies. All experiments used the same *Stop condition*, which is met when:

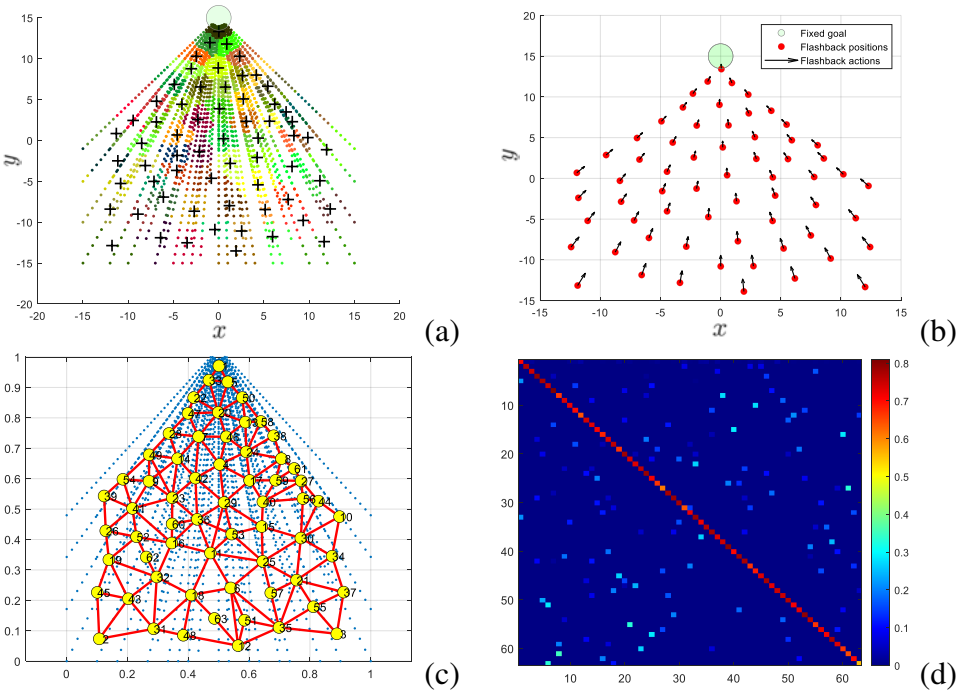


Fig. 3.3 Learning reference model (flash-back memory). a) clustering of GEs, b) mean velocity of each cluster, c) clusters' relationship, and d) generated transition matrix.

- a minimum distance to the target is accomplished (success),
- a maximum navigation time in the environment is reached (lost),
- the agent goes out of boundary (outside).

We evaluate the performance of the proposed method and compare it with three learning algorithms, namely, the general cumulative reward-based Q-learning [143], inverse reinforcement learning (IRL) [101], and self-learning (SL) in the RL context (distance-based evaluation).

3.3.4 Performance Evaluation

The action selection procedure has a big impact on the L's effort to reach the targeted G. A good policy requires fewer actions and, in parallel, less time to finish the mission. Fig. 3.4 shows the number of actions taken by L for each episode using different methods. It describes that the presented online incremental IL model (OIL) makes fewer actions than other methods. This can be explained by the evaluation of L's movement using $\boldsymbol{D}$ can improve the actions that lead the agent to the desired next state. L adopting the proposed method has higher successful trajectories than SL, Q-learning, and IRL, as depicted in Fig. 3.5. Our approach

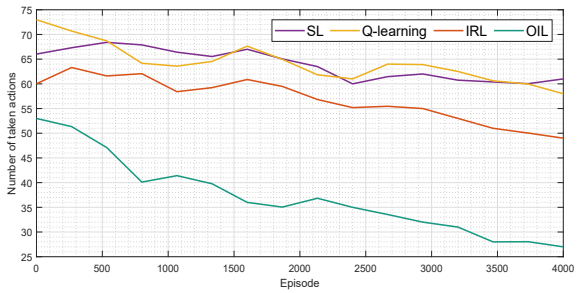


Fig. 3.4 The number of performed actions by the agent during each episode.

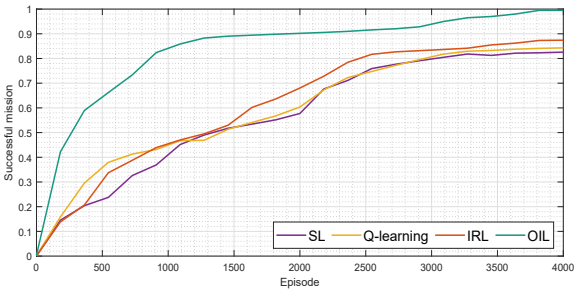


Fig. 3.5 The success rate to reach the target G in each episode.

uses a threshold ρ to initialize learning of new explored s^L in Q-table (see (3.12)). Since ρ has a great impact on the Q-function's complexity, we train L with different ρ values, depending on the distance between the current state s_t^L and the set of recorded states (S_Q) in Q-table. By considering the success rate and the required execution time obtained by each ρ value, we select the suitable value as shown in Table 3.1. $\rho = 1$ and $\rho = 3$ have almost the same success rate but the required time by $\rho = 3$ is more optimal than $\rho = 1$.

Table 3.1 Training the learning model with different ρ values. The selected threshold is $\rho = 3$.

ρ	1	1.5	2	2.5	3	3.5	4	4.5	5
success (%)	96.74	95.99	96.11	96.04	96.52	94.87	93.01	91.56	88.93
time (s)	110.49	93.71	102.02	99.89	90.24	97.83	100.04	109.33	114.26

Moreover, Fig. 3.6 demonstrates how modifying the action selection procedure can reduce the exploration and minimize the imitation cost resulting in a high learning rate during the training phase.

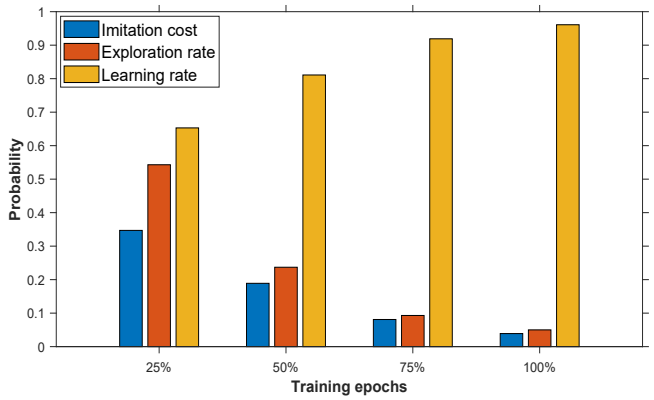


Fig. 3.6 Presenting the exploration and learning rates after each training quarter and their impact on the imitation cost.

3.3.5 Learning Cost Evaluation

Two main factors affect $\mathscr{G}$ (see (3.10)), the action difference at time k (see (3.7)) and the state divergence after performing a_k by L (see (3.9)). Fig. 4.29 illustrates the imitation loss in both policies ρ_a where L is under control of action selection at each time instant k, and policy ρ_s which by improving a_k^L leads to minimizing the divergence between prediction and evidence. Further, Fig. 4.29 shows that $\mathscr{G}$ drops down capably in less than 2k training episodes, and after 3k episodes, its value tends to stable below 0.1, reaching about 0.039. Therefore, L learns to maximize the likelihood with $\boldsymbol{D}$. Fig. 3.8-(a)-(b) presents the performance of the

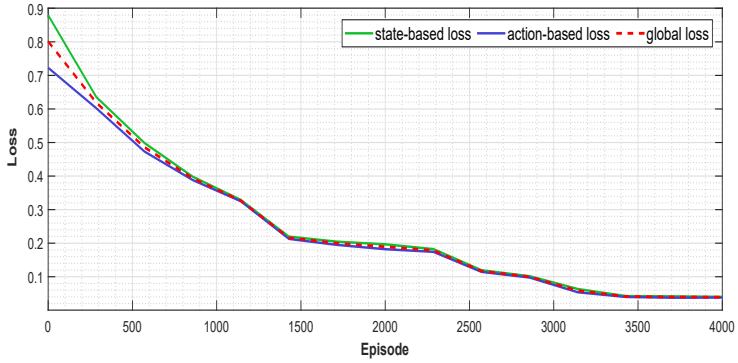
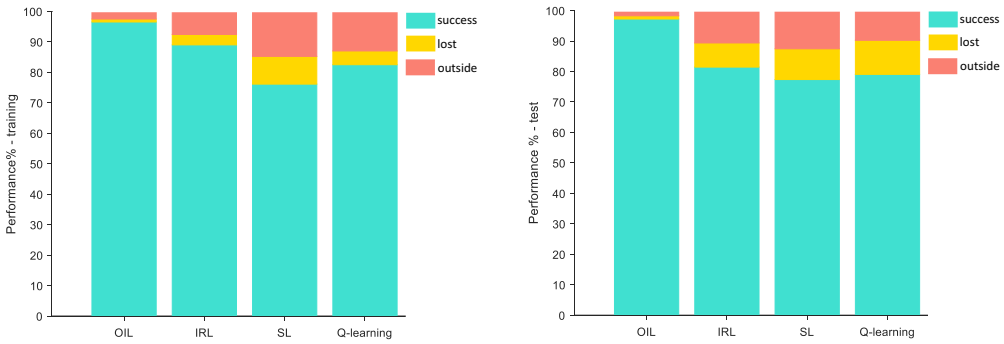


Fig. 3.7 Imitation loss measurements individually and global.

proposed method during training and testing, respectively, in terms of success, lost, and going outside (explained in 3.3.3). Also, Fig. 3.8-(a)-(b) provides a comparison with other learning methods. It is demonstrated that the proposed method (OIL) outperforms others in both the training and testing stages, which is attributed to the effectiveness of motion prediction while

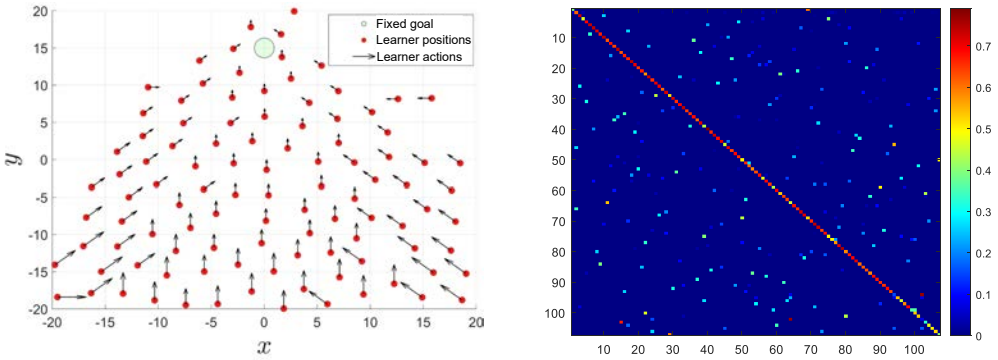
dealing with abnormalities that improve the success rate. Additionally, during testing, results showed that by $4k$ training episodes, L could move in a continuous environment to effectively reach G whereas other methods still have a high failure rate.



(a) (b)

Fig. 3.8 Results after $4k$ training episodes. a) Training results, and b) Testing results. In both stages, OIL has higher success than other methods.

As discussed in subsection 3.2.3, L clusters all observed states and the corresponding performed actions. The recorded pairs are clustered to calculate the mean action value of the corresponding clusters for having comparable data with $\boldsymbol{D}$ and avoid looking in too many states in Q-table. Fig. 3.9-(a)-(b) depict the Q^* clusters (see (3.13)) and the corresponding updated transition matrix.

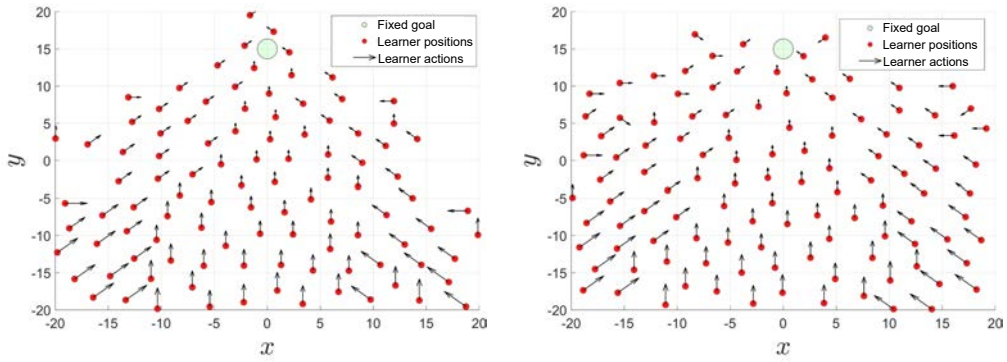


(a) (b)

Fig. 3.9 Discrete state-action representation from global imitation policy $\mathscr{G}$ (a), and the novel generated transition matrix (b).

Comparing sub-figures (Fig. 3.3-(d) and Fig. 3.9-(b)) show that L has an expanded Π than $\boldsymbol{D}$ after exploring and learning new states, allowing L to predict better performance and

select desired actions. Under probabilistic inference, such an incremental learning process endows L with the capability of avoiding abnormal states. Meanwhile, to evaluate the effect of each imitation strategy (ρ_a and ρ_s), L is trained with each policy individually. Fig. 3.10-(a)-(b) demonstrate the learned clusters through each policy. Comparing Fig. 3.9-(a) with Fig. 3.10-(a)-(b) shows how applying global policy ($\mathscr{G}$) generates the most efficient training.



(a) (b)

Fig. 3.10 Discrete state-action representation from the action-based policy ρ_a (a), and the state-based policy ρ_s (b).

In the testing stage, three Q*-tables obtained from $\mathscr{G}$, ρ_a and ρ_s are employed with ε-greedy $= 0$ to generate 300 trajectories (from new start positions rather than training stage) that are compared with the E's behavior. We use two distance measurements to compare trajectories from testing the provided Q*-tables with the E's behaviors: Spatio-Temporal Euclidean Distance (STED) [100] and Symmetrized Segment-Path Distance (SSPD) [16]. STED uses temporal information by comparing trajectories point to point. SSPD is a shape-based distance that compares trajectories as a whole. Table 3.2 shows the mean value of distance measurements between test trajectories (over 300 starting points) of different imitation policies and E's behaviors. Furthermore, Table 3.2 presents the quantitative results from testing the trained models by $\mathscr{G}$, ρ_a and ρ_s in terms of success, lost, and going outside (explained in 3.3.3).

Table 3.2 Testing results after $4k$ training episodes.

imitation policy	success (%)	lost (%)	outside (%)	STED	SSPD
trained by $\mathscr{G}$	97.22	1.05	1.73	0.551	0.164
trained by ρ_a	90.76	4.23	5.01	1.105	0.362
trained by ρ_s	89.01	4.96	6.03	1.359	0.411

3.4 Model II - Multi-Agent Dynamic Interaction

The integration of both modalities, RL and IL, enable the learning of complex skills from raw sensory observation[96]. This section proposes a framework integrating RL and IL for tracking a dynamic target. Moreover, it allows adapting to perturbation in a dynamic environment. IL is used as a pre-training step to encode the expert demonstrations in a coupled GDBN (C-GDBN) for a specific task (i.e., reaching a dynamic target). The C-GDBN is a probabilistic graphical model explaining the dynamic interactions among multiple environmental agents. Accordingly, the sub-optimal demonstrations can be explained by a set of configurations between the expert agent (E) and a dynamic target (T). Therefore, the model is able to explain the interaction between IA and its surroundings.

3.4.1 Learn a Coupled Generalized Dynamic Bayesian Network

This section presents a dynamic interaction model based on the E behavior during an offline learning phase. The aim of the offline learning phase is to learn a reference model (RM) that the learning agent (L) can use for initializing the online learning model. Initialization is conducted by mapping the reference C-GDBN structure onto the L moving reference system as a reference model.

The RM consists of a C- DBN [9] representing the interaction of two dynamic entities, E and T. The model is described by means of a set of observation and state variables that describe the state of the two interacting agents at a given time instant k. It is assumed that the moving agents' (E and T) observations are represented by variables $\mathrm{Z}_k^{\mathrm{E}}$ and $\mathrm{Z}_k^{\mathrm{T}}$, respectively (① in Fig. 3.11).

At a higher level, hidden continuous GSs [56] can be formed describing the agents' instantaneous dynamics up to a chosen n-th order temporal derivative. Thus, a joint GS ($\tilde{\mathrm{X}}_k$) (② in Fig. 3.11) incorporating the dynamics of multiple agents (i.e., E and T) at each time instant k can be defined as follows:

$$\tilde{\mathrm{X}}_k = [\tilde{\mathrm{X}}_k^{\mathrm{E}} \ \tilde{\mathrm{X}}_k^{\mathrm{T}}]^{\intercal}, \tag{3.16}$$

where $\tilde{\mathrm{X}}_k^{\mathrm{E}}$ and $\tilde{\mathrm{X}}_k^{\mathrm{T}}$ denote the GSs of E and T, respectively. Here, a GS related to agent i (i.e., $\tilde{\mathrm{X}}_k^i$) is defined as a vector composed of the agent's state and its first-order temporal derivative, such that $\tilde{\mathrm{X}}_k^i = [x \ \dot{x}]^{\intercal}$ where $x \in \mathbb{R}^d$, $\dot{x} \in \mathbb{R}^d$, $i \in \{\mathrm{E}, \mathrm{T}\}$ and d stands for the dimensionality of the state vector. Each observed sensor variable Z_k^i is assumed to be related to the corresponding agent's hidden state variable $\tilde{\mathrm{X}}_k^i$ by a linear relationship according to

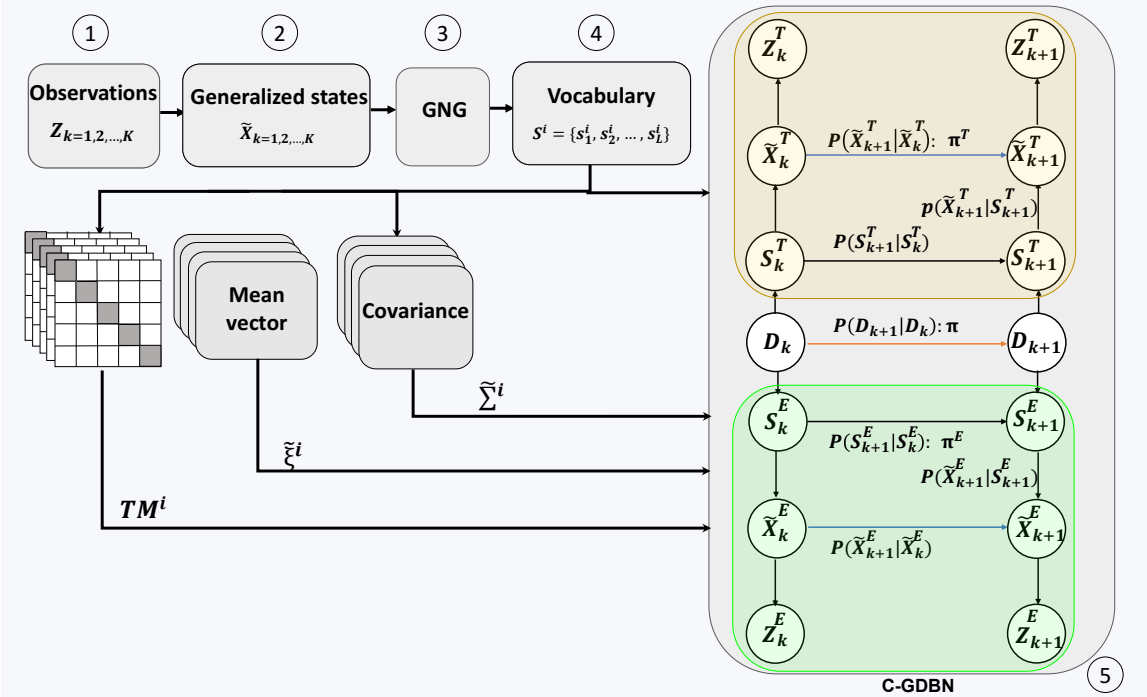


Fig. 3.11 An overview of learning a Dynamic Interaction Model. The arrows in c-GDBN represent the conditional probabilities between involved variables. Vertical arrows facilitated the causalities description between continuous and discrete levels of inference and observed measurements. Horizontal arrows explain temporal causalities between hidden variables. In particular, the orange arrow encodes the interaction of couples of agents, and the blue arrows represent the influence at a continuous level.

the following observation model:

$$\mathrm{Z}_k^i = \mathrm{H}\tilde{\mathrm{X}}_k^i + v_k, \tag{3.17}$$

where $H = [\mathrm{I}_d \;\; 0_{d,d}]$ is the observation matrix that maps hidden GSs ($\tilde{\mathrm{X}}_k^i$) to measurements (Z_k^i) and v_k is the measurement noise which is assumed to be zero-mean Gaussian with covariance R, such that, $v_k \sim \mathcal{N}(0, \mathrm{R})$.

To learn the dynamic interaction models, we first assumed that there is no external force influencing the evolution of GSs of the observed agents under the static equilibrium assumption described by the following model:

$$\tilde{\mathrm{X}}_k^i = \mathrm{A}\tilde{\mathrm{X}}_{k-1}^i + w_k, \tag{3.18}$$

where $\mathrm{A} \in \mathbb{R}^{d \times d}$ is the dynamic matrix and w_k is the process noise which is assumed to be a zero-mean Gaussian with covariance Q, such that $w_k \sim \mathcal{N}(0, \mathrm{Q})$.

This implies a null acceleration, and the learning approach consists of observing deviations from such hypothesized equilibrium through an active approach (i.e., NFF). An NFF can be interpreted as a generalized KF, which uses the innovations obtained by observing an input data sequence Z_k^i to estimate a new expert demonstration that describes interactions between observed agents in the GS space.

The innovations can be seen as mismatches between observations (obtained by observing interaction) and predictions (based on the assumption that the observations should be quasi-static) defined as follows:

$$\dot{\upsilon} = \mathrm{H}^{-1}\left(\mathrm{Z}_k^i - \mathrm{H}\tilde{\mathrm{X}}_k^i\right), \tag{3.19}$$

the couples $(\tilde{\mathrm{X}}^i, \dot{\upsilon})$ obtained by NFF along the interaction time series are defined as GEs. Those GEs can be clustered using an unsupervised method. We employ the Growing Neural Gas with utility measurement (GNG-U) [73], which outputs a set S^i of (switching) discrete variables (i.e., clusters) representing the discrete level of the C-GDBN (⑤ in Fig. 3.11). Each cluster describes in which region of the GS space, with which difference in the dynamic motion (w.r.t the hypothesized absence of external forces), and at what time a specific interaction has occurred.

The joint vocabularies of switching variables from agents' GEs, E and T, describe a specific type of interaction among the agents at multiple levels (i.e., discrete and continuous levels). Each discrete state represents a region where quasi-linear models are valid to present the interactive dynamical system over time. Vocabularies are defined as:

$$\mathrm{S}^i = \{\mathrm{s}_1^i, \mathrm{s}_2^i, \ldots, \mathrm{s}_{\mathscr{L}_i}^i\}, \tag{3.20}$$

where $\mathscr{L}_i$ is the total number of clusters associated with agent i and $\mathrm{s}_l^i \in \mathrm{S}^i$ is a specific cluster describing agent's motion.

Since each superstate s^i is supposed to follow a multivariate Gaussian distribution, it can be represented by its sufficient statistics, specifically, the covariance matrix $\tilde{\Sigma}_{\mathrm{s}_k^i}$ and the generalized mean values $\tilde{\mu}^{\mathrm{s}^i} = [\mu_{Pos}^{s^i}\ \mu_V^{s^i}]$, where $\mu_{Pos}^{s^i}$ and $\mu_V^{s^i}$ represent the mean value of the states (on position) and the mean value of the corresponding derivatives (on velocity), respectively.

In a time instant k, each agent i is represented by an active superstate $s_k^i \in S^i$. Joint active superstates from different agents occurring simultaneously form an interaction configuration defined as $\mathrm{D}_k = [\mathrm{s}_k^{\mathrm{E}}, \mathrm{s}_k^{\mathrm{T}}]^\intercal$. Consequently, an additional vocabulary of dictionary configurations can be defined and included in the C-GDBN at a higher hierarchical level, such that:

$$\mathrm{D} = \{\mathrm{D}_1, \mathrm{D}_2, \cdots, \mathrm{D}_{\mathrm{M}}\}, \tag{3.21}$$

where M is the total number of configurations and $\mathrm{D}_m \in \mathrm{D}$ encodes a given identified configuration composed of the position and velocity features of both agents and defined as:

$$\mathrm{D}_m = \left[(\mu_{Pos}, \mu_V)^{\mathrm{E}}, (\mu_{Pos}, \mu_V)^{\mathrm{T}}\right]. \tag{3.22}$$

The inter-slice links at multiple levels among consecutive time instants are also learned to define the DBN completely. It has to be noted that the learned switching variables are associated with corresponding dynamic models at the GS continuous level. As the NFF clusters similar innovations into compact regions of the state space, in each region, it is possible to estimate the interaction force for a given agent by modifying the dynamic model of (3.17). Regarding linearity and Gaussianity of the NFF dynamic model, the dynamic model of each agent inside a cluster s^i is estimated based on the quasi-constant velocity that depends on the state and derivative mean values of GEs clustered in each s^i, such that:

$$\tilde{\mathrm{X}}_k^i = \mathrm{A}\tilde{\mathrm{X}}_{k-1}^i + \mathrm{B}\mu_V^{s_k^i} + w_k, \tag{3.23}$$

where $\mathrm{B} \in \mathbb{R}^{d \times d}$ is a control model matrix that maps the agent's velocity estimation into the following states. The variable $\mu_V^{\mathrm{s}_k^i}$ is a control vector encoding the agent's motion when it is found in a region s_k^i that can be formulated as:

$$\mu_V^{s_k^i} = [\dot{x}_{s_k^i}, \dot{y}_{s_k^i}], \tag{3.24}$$

where $\dot{x}_{\mathrm{s}_k^i}$ and $\dot{y}_{\mathrm{s}_k^i}$ are the velocity components of agent i associated with s_k^i. The transition model defined in (3.23) corresponds to cluster-dependent motivated dynamics whose effects are encoded in $\mu_V^{\mathrm{s}_k^i}$ and switched according to the activated configuration. The probabilistic law that regulates switching among different local forces captured by different interaction configurations can be estimated in different ways (e.g., frequentist or geometrical) and encoded in a Transition Matrix (Π). Learning the Π involves estimating the transition probabilities $\mathrm{P}(\mathrm{D}_{k+1}|\mathrm{D}_k)$ of switching from a current configuration (D_k) to another one (D_{k+1}) and it is defined as:

$$\Pi = \begin{bmatrix} \mathrm{P}(\mathrm{D}_1|\mathrm{D}_1), & \mathrm{P}(\mathrm{D}_1|\mathrm{D}_2), & \ldots, & \mathrm{P}(\mathrm{D}_1|\mathrm{D}_\mathrm{M}) \\ \mathrm{P}(\mathrm{D}_2|\mathrm{D}_1), & \mathrm{P}(\mathrm{D}_2|\mathrm{D}_2), & \ldots, & \mathrm{P}(\mathrm{D}_2|\mathrm{D}_\mathrm{M}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{P}(\mathrm{D}_\mathrm{M}|\mathrm{D}_1), & \mathrm{P}(\mathrm{D}_\mathrm{M}|\mathrm{D}_2), & \ldots, & \mathrm{P}(\mathrm{D}_\mathrm{M}|\mathrm{D}_\mathrm{M}) \end{bmatrix}, \tag{3.25}$$

where $\sum_{m}^{\mathrm{M}} \mathrm{P}(\mathrm{D}_p|\mathrm{D}_m) = 1$ such that $p, m \in \mathrm{M}$.

3.4.2 Initialize the Learning Model

The learning model can be interpreted as an RM transformed in such a way that allows L directly uses its own observations and generate state series describing its relative state with respect to another interacting dynamic agent (i.e., T). It provides L with the capability to imitate the expert motions by generating transformed sequences from the RM. L considers C-GDBN nodes to initialize the learning model, which can be used to predict interaction states under the perspective of a learning agent. Accordingly, L exploits the generative C-GDBN in terms of pure IL to initialize a generalized DBN considering the interactive behavior of the learning agent with its surroundings (see Fig. 3.12).

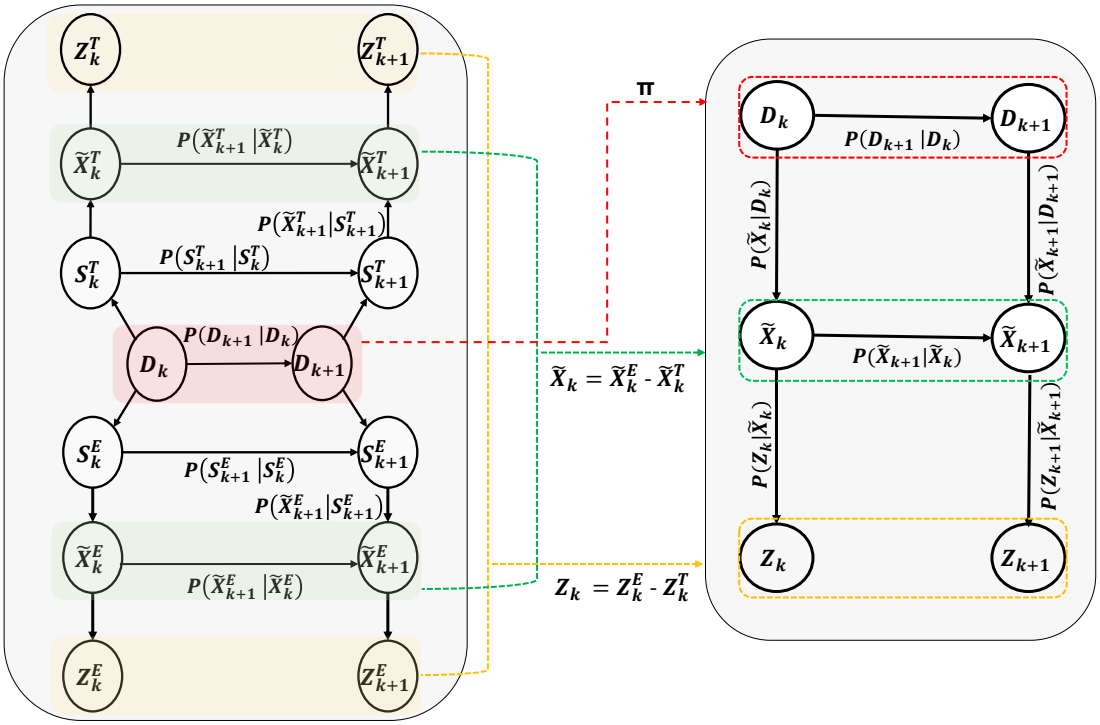


Fig. 3.12 Initializing the the learning model model (right side) by exploiting the reference model (left side). The reference model shows the C-GDBN from Fig. reffig1-system. The learning model's arrows represent conditional probabilities such as, vertical arrows introduce causalities between both (discrete and continuous) levels of influence and observed measurements. Horizontal arrows explain temporal causalities between hidden variables.

The continuous level (CL) of RM is employed to provide the generalized relative distance between E and T, which consists of the relative distance and the velocity. The generalized

relative distance can be seen as the difference of joint GSs describing the interaction at the continuous level of the two agents in a specific configuration (D_m) and defined as:

$$\tilde{\mathrm{X}}_k = \left[\tilde{\mathrm{X}}_k^{\mathrm{E}} - \tilde{\mathrm{X}}_k^{\mathrm{T}}\right] = \left[(x_k^{\mathrm{T}} - x_k^{\mathrm{E}}), (\dot{x}_k^{\mathrm{T}} - \dot{x}_k^{\mathrm{E}})\right]. \tag{3.26}$$

The relative distance vector is highlighted as the difference in absolute coordinates and velocities of the two agents, that in the online learning phase, it will be calculated as the relative distance between L and T.

The discrete level (DL) of the RM is employed to represent the learned set of configurations $\mathrm{D}_m \in \mathrm{D}$. During the online learning phase, L is assumed to take the role of E. Therefore, all clusters related to E should correspond to the clusters describing L states in a certain configuration.

By providing a biunivocal mapping between clusters of E and L, the transition probabilities composing the Π model can characterize the temporal dependencies of discrete series of interactions in the learning model among L and T for the experiences to be imitated. Thus, the transition model can be directly mapped onto the online learning model from the corresponding Π of the RM (i.e., $\Pi^{\mathrm{L}} = \Pi^{\mathrm{E}}$). Moreover, the observation (Z_k) that represent the generalized relative distance between L and T can be mapped onto the RM according to:

$$\mathrm{Z}_k = \left[\mathrm{Z}_k^{\mathrm{E}} - \mathrm{Z}_k^{\mathrm{T}}\right] \tag{3.27}$$

To this end, a configuration $\mathrm{D}_m \in \mathrm{D}$ at the DL of the learning model is represented by a joint superstate of each agent at time instant k, i.e., $\mathrm{D}_k = \left[s_k^{\mathrm{L}}, s_k^{\mathrm{T}}\right]$. Thus, the model can predict the expected future configurations based on the dynamic transition rules encoded in the transition matrix Π^{L}. Therefore, the Q function is initialized as:

$$\mathrm{Q} = \begin{bmatrix} \mathrm{P}(a_1|D_1) & \mathrm{P}(a_1|D_2) \ldots \mathrm{P}(a_1|D_M) \\ \mathrm{P}(a_2|D_1) & \mathrm{P}(a_2|D_2) \ldots \mathrm{P}(a_2|D_M) \\ \vdots & \vdots \quad \ddots \quad \vdots \\ \mathrm{P}(a_{\mathrm{N}}|D_1) & \mathrm{P}(a_{\mathrm{N}}|D_2) \ldots \mathrm{P}(a_{\mathrm{N}}|D_M) \end{bmatrix}, \tag{3.28}$$

where where $\sum_n^{\mathrm{N}} \mathrm{P}(a_n|\mathrm{D}_m) = 1$ such that $m \in \mathrm{M}$, $n \in \mathrm{N}$, and $a \in \mathscr{A}$ in a set $\mathscr{C}$. Q is an incremental function that will be modified and developed by novel experiences during the online learning phase.

3.4.3 Online Abnormality Measurement

The objective online learning stage is to learn imitation policies by minimizing the abnormalities during tracking a dynamic target. The proposed incremental learning model takes advantage of the RL context to learn multiple imitation policies and regulate the L's movements to accomplish its task. The model estimates the activated configuration $(\dot{\mathrm{D}}_k)$ at each time instant to evaluate the L's behavior by calculating the abnormalities at both continuous and discrete levels of the GDBN. $\dot{\mathrm{D}}_k$ is the closest learned configuration to the current L's configuration (D_k^L) measured by Euclidean distance. The learning model uses the likelihood estimation between the active configuration and the current observation from the learning agent to estimate the novelty of the current configuration D^L. The determined prior by the hidden states and actions at the previous time instant can affect the L's decision-making.

The online learning model evaluates the validity of the L's current configuration using the divergence between the observation and the expectation. L employs Kullback Leibler-Divergence $(\mathscr{D}_{\mathscr{KL}})$ [83] between the measured relative distance by L at time instant k $(\tilde{X}_k^L|D_k^L)$ and the corresponding relative distance to the active configuration $(\tilde{X}_k^{\dot{D}}|\dot{D}_k)$ to measure the abnormality at the CL of GDBN after each performed action (a_{k-1}), as:

$$\rho_{CL,k} = \mathscr{D}_{\mathscr{KL}}\left(\tilde{\mathrm{X}}_k^L||\tilde{X}_k^{\dot{D}}\right) = \int \tilde{\mathrm{X}}_k^L \log\left(\frac{\tilde{\mathrm{X}}_k^L}{\tilde{X}_k^{\dot{D}}}\right)\mathrm{d}\tilde{\mathrm{X}}_k. \tag{3.29}$$

3.4.4 Action Selection and Update the Model

The action selection is based on two parameters, namely, the normalized measured abnormality $\mathscr{G}$ at time k, which the normalized $\rho_{CL,k}$ (defined in 3.29), where ρ_{CL} measures the likelihood between the current L configuration and reference one, and a threshold ($t \in [0,1]$), which is based on a trial and error process. Therefore if $\mathscr{G}$ becomes very low, which means that L follows the expectation, it can exploit the preferred action associated with the active configuration $(a_k|\dot{D}_k)$. On the other hand, if the $\mathscr{G}$ amount appears high, it is required to generate a random action. L uses t to whether to exploit or explore an action as:

$$a_k \sim \begin{cases} \underset{a_k}{\arg\max}\ \mathrm{Q}(\mathscr{A}, \dot{\mathrm{D}}_k), \text{ if } \mathscr{G}_k < t \ \text{ (exploitation)}, \\ \textit{random} \ \text{ from } \mathscr{A}, \text{ if } \mathscr{G}_k \geq t \text{ (exploration)}, \end{cases} \tag{3.30}$$

where $\mathscr{A} = \{a_1, a_2, \ldots, a_8\}$ is a set of eight cardinal and ordinal directions[2]

[2]The 8 directions are North, South, East, West, North-West, North-East, South-East, South-West.

L records the explored configuration (D_k^+) along with the performed actions ($a_k \in \mathscr{A}$) in a incremental function $Q(D,a)$ and the novel configurations are saved in a set D_Q that grows incrementally as experiences are observed over time:

$$\mathrm{Q} = \begin{bmatrix} \mathrm{P}(a_1|\mathrm{D}_1) \dots \mathrm{P}(a_1|\mathrm{D}^+) & \dots \\ \mathrm{P}(a_2|\mathrm{D}_1) \dots \mathrm{P}(a_2|\mathrm{D}^+) & \dots \\ \vdots \quad \ddots \quad \vdots & \vdots \\ \mathrm{P}(a_\mathrm{N}|\mathrm{D}_1) \dots \mathrm{P}(a_\mathrm{N}|\mathrm{D}^+) & \dots \end{bmatrix}, \tag{3.31}$$

where $\sum_{n=1}^{\mathrm{N}=8} \mathrm{P}(a_n|\mathrm{D}_m^+) = 1$ such that D_m^+ are the new explored states. In order to weigh up the trained model than *RM*, L clusters all the recorded pairs $[\mathrm{D}_k^+, a_k]$ by employing GNG. The latter outputs a set of clusters representing the new configurations ($\hat{\mathrm{D}}$) and the corresponded mean actions (ȧ) which are added to the updated Q-table (Q^*) defined as:

$$\mathrm{Q}^* = \begin{bmatrix} \mathrm{P}(\dot{a}_1|\hat{\mathrm{D}}_1) & (\dot{a}_1|\hat{\mathrm{D}}_2) \dots \mathrm{P}(\dot{a}_1|\hat{\mathrm{D}}_M) \\ \mathrm{P}(\dot{a}_2|\hat{\mathrm{D}}_1) & (\dot{a}_2|\hat{\mathrm{D}}_2) \dots \mathrm{P}(\dot{a}_2|\hat{\mathrm{D}}_M) \\ \vdots & \vdots \quad \ddots \quad \vdots \\ \mathrm{P}(\dot{a}_N|\hat{\mathrm{D}}_1) & (\dot{a}_N|\hat{\mathrm{D}}_2) \dots \mathrm{P}(\dot{a}_N|\hat{\mathrm{D}}_M) \end{bmatrix}. \tag{3.32}$$

L adapts the action selection procedure by updating the Q-table defined in (3.31) based on the abnormality measurement at each time instant k. Since the provided Q is a probabilistic table, updating the Q value can be rewritten in a probabilistic form as follows:

$$\mathrm{Q} = (1 - \eta)\mathrm{P}(a_{k-1}|\mathrm{D}_{k-1}) + \eta\Big[(1 - \mathscr{G}_k) + \gamma \max_{a_k} \mathrm{P}(a_k|D_k)\Big], \tag{3.33}$$

where η is the learning rate that controls how quickly the learning agent adopts to the explorations imposed by the environment, $(1-\mathscr{G}_k)$ is the normalized reward measurement with a range in $[0,1]$, and γ is a discount factor.

3.5 Simulation and Performance Evaluation - Model II

3.5.1 Experimental Setup

In this section, we provide numerical results to validate the proposed method. We consider a table of trained data where L chases the target G in a 40×40 space (see Fig.3.13). In training data, L's motion is described by 8 different motion unit-vectors associated with the cardinal and intercardinal directions. G motions consists in a horizontal dynamics along the x axis at a

fixed height point y_G. Accordingly, G can move in two senses: right or left inside the interval $[x_G^{(min)}, x_G^{(max)}]$. G dynamics consists of a continuous motion in one sense until it reaches an interval boundary. Then, it starts moving in the opposite sense covering only the defined interval points. The speed of G movement is different than the expert experiments in the RM to guarantee that L learns to reach the target in a new scenario. The following parameters are employed for simulation purposes: $y_G = 15$, $x_G^{(min)} = -15$ and $x_G^{(max)} = 15$.

The experiments are executed in a simulated environment through 500 episodes with different start positions to train a learning-agent L. Each episode consists of 10 iterations, i.e., L tries 5k iterations by 500 different start positions to learn the policies. We evaluate the performance of the proposed framework and compare it with other learning algorithms from the literature, namely, the general Q-learning [143] and double Q-network [67]. Results related to the capabilities of detecting abnormalities and evaluating the current model are explained in detail as follows.

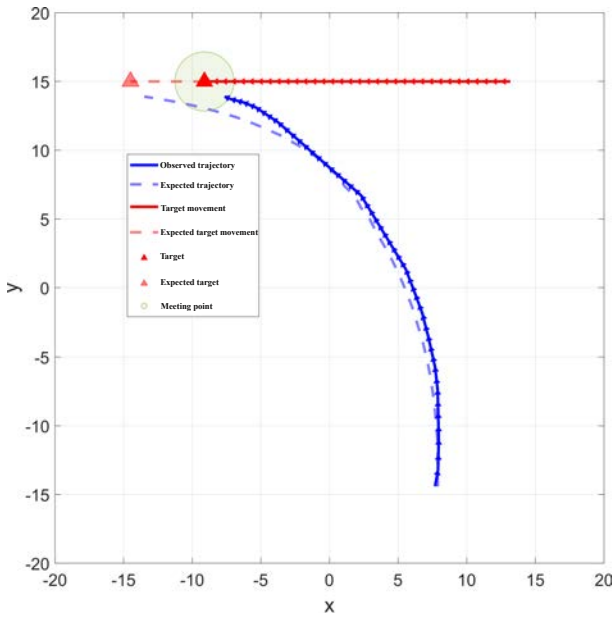


Fig. 3.13 Example of matched trajectories.

3.5.2 Performance Evaluation

After the trial stage, L acquires knowledge about the contingencies, and the likelihood mapping in the generative model is aligned adequately with the reference generative process and the targeted goal (i.e., reaching a dynamic target). Crucially, we assume that the correctness and accuracy of the action selection procedure guide the learning agent to the

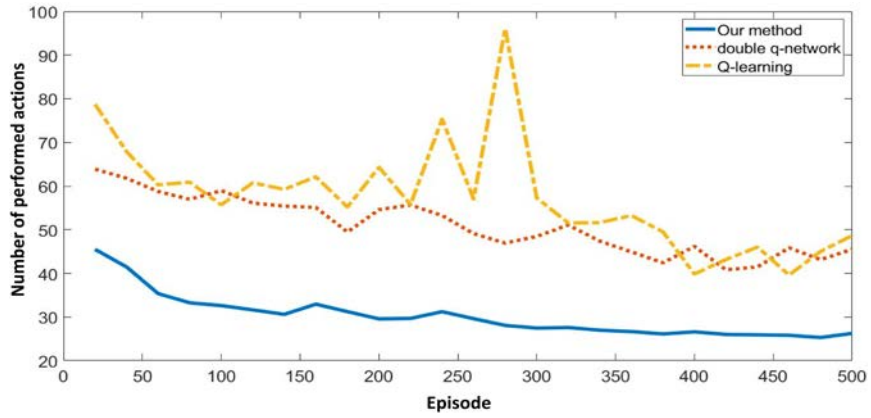


Fig. 3.14 Number of performed actions in each training episode.

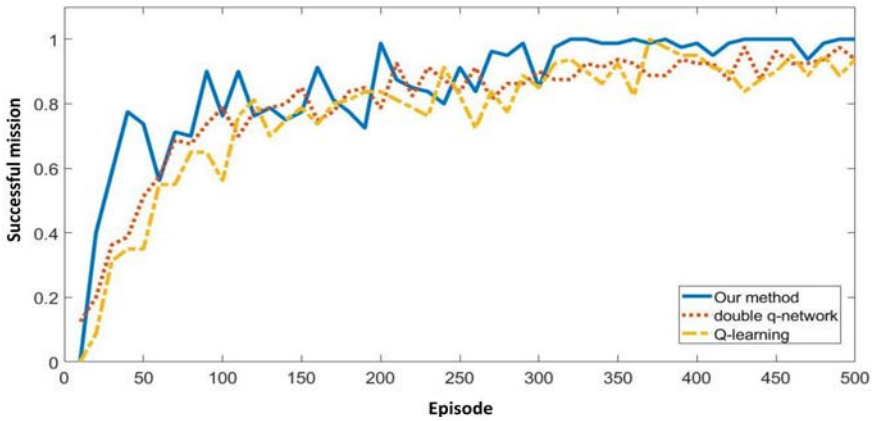


Fig. 3.15 The gained success rate in each training episode.

expected observations. Fig. 3.14 illustrates that L movements are engaged coherently, which causes less exploration in each trial epoch (e.g., each episode).

Additionally, Fig. 3.14 compares the number of executed actions during the training using different learning methods, where it shows L performs fewer actions to accomplish its task by using our method than others. Moreover, L adopting the proposed method has higher successful trajectories than other methods, as depicted in Fig.3.15.

3.5.3 Learning Cost Evaluation

Evaluating the current model's configurations during the online learning phase is employed to detect abnormalities. Fig. 3.16 shows abnormality estimation based on the distinction between the current observation and the RM's prediction after performing an action through $\mathscr{D}_K L$ measurements (see 3.29). The result demonstrates that the high abnormality values

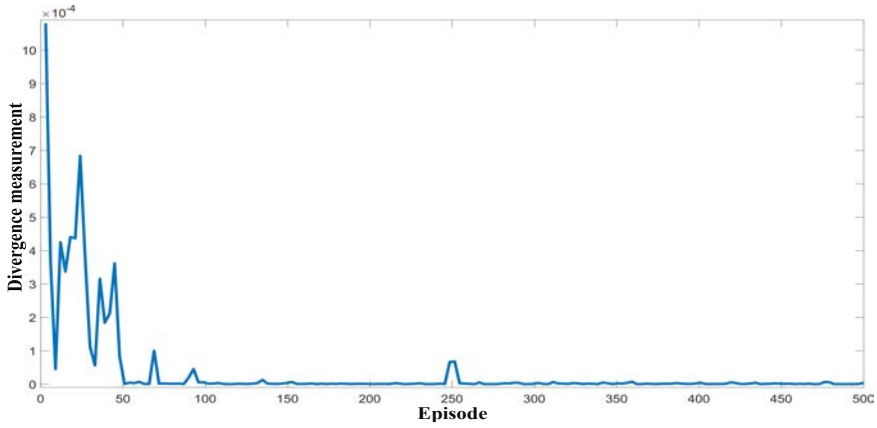


Fig. 3.16 Abnormality measurement.

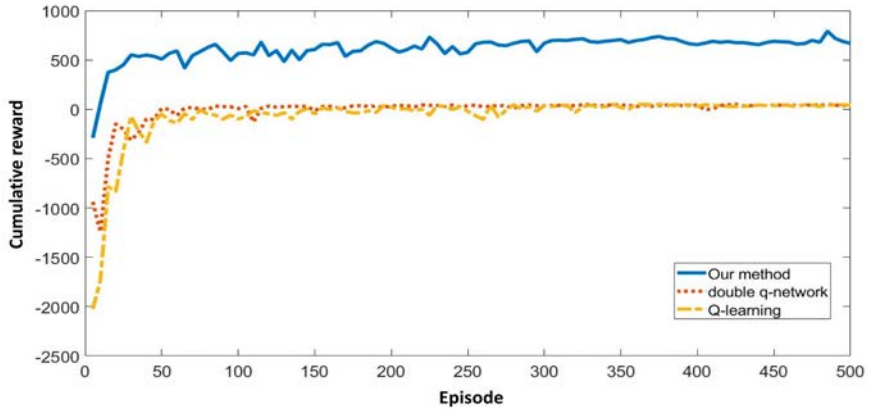


Fig. 3.17 Cumulative reward during the trial.

are present in the learning's initial portion, and Once L learns the reward policies, the measurements go down dramatically.

Modifying the actions selection procedure can minimize imitation costs, resulting in a high learning rate during the training phase. Our goal is to find the best set of actions that minimize imitation loss (maximizing the reward). Fig. 3.17 demonstrates the cumulative reward during 5k training iterations. As the results show, the learner agent by employing the proposed model, succeeds in imitating the expert demonstration rather than other learning methods.

3.6 Conclusion

This chapter introduces two system models using Generalized Dynamic Bayesian Networks, namely Model I, which is based on a single dynamic agent, and Model II, which considers

multi-agent dynamic interaction. Model I proposed an incremental imitation learning model where an intelligent agent tracks a stationary target. The imitator learns the interaction with surroundings by observing an expert agent. This section develops a probabilistic model where the learning agent does not require explicitly repeating the expert agent's behaviors. Therefore, the learner is not limited to recalling exact observations of the optimal behavior but employs a probabilistic model as a flashback memory for guiding a reinforcement learning approach that allows the learning agent to learn the previous experiences on its own.

Model II proposed an adaptive probabilistic model for imitation learning in a dynamic environment. In this model, imitation learning is used as a pre-training step to encode the expert demonstrations in a coupled Generalized Dynamic Bayesian Network for reaching a non-stationary target which enables the learning agent to take uncertainty appropriately into account. The presented method demonstrates learning from a dynamic interaction model to minimize the cost of imitation during the online learning phase.

In both system models, experimental results show the capability to minimize the abnormalities while learning the policies from the sub-optimal demonstrations. Those abnormalities can be used as qualitative observation in order to learn from unseen situations.

Chapter 4

Active Inference for Incremental Imitative Learning in Autonomous Driving

4.1 Introduction

Autonomous driving (AD) requires the resolution of perception and motion planning issues in the presence of dynamic objects interacting with the environment. The complex interactions between multiple agents are significant challenges due to the difficulty of predicting their future motions. Most model-based AD approaches necessitate designing the driving policy model manually [109, 65]. While designing a decision and planning system for AD is complex, an alternative is to learn the driving policy from an expert agent (E) using imitation learning (IL) [71]. Existing IL approaches can handle simple driving tasks such as lane following. However, if the agent is dealing with a new environment or a more complicated task (e.g., lane-changing), it is required that the human driver has to take control, or the system fails ultimately [20, 127].

Modifying a learning agent's (L) actions to lead to the prior preference for future observation is an effective mechanism to adapt to environmental changes. Active Inference (AIn) [57, 51] suggests a framework where the agent chooses actions that minimize the expected surprise (abnormality) and improve the description of how the agent expects itself to behave. Surprise is the divergence between expectation and evidence, and it is an information-theoretic quantity that can be approximated with variational Free Energy (FE) [53] and can be treated as a negative value function (e.g., imitation loss) to optimize the decision-making in autonomous systems. FE explains perception, action, and model learning

in a Bayesian probabilistic way that provides an upper bound on the negative log-evidence or surprise [47]. This chapter claims that AIn can be interpreted using reinforcement learning (RL) algorithms by learning a preferred observation from an expert and realizing a theoretical relation between them.

In this work, IL is used as a pre-training step to learn the dynamic interaction between an expert and a moving object during a specific task in an unsupervised manner (e.g., an expert overtakes a dynamic object). The dynamic interaction (or expert demonstration) can be represented in coupled Generalized Dynamic Bayesian Networks (C-GDBNs) that express both hierarchical and temporal relationships among high-level variables capturing semantic information and low-level variables capturing rough sensory information. However, in IL, E demonstrates only the optimal policy (allowing L to follow the optimal trajectory), and if L deviates from that policy even slightly, it will be unable to recover since training and testing are sampled from different distributions. Thus, it will suffer from the well-known problem of distributional shift [40]. To overcome such an issue, we propose integrating AIn with IL, allowing L to predict the E's behavior and evaluate the encountered situation. If predictions match observations, L selects the same actions performed by E (i.e., pure imitation learning). Otherwise, if L deviates from the E's predicted trajectory, it starts exploring new actions allowing L to recover by moving toward the expert reference model. Thus, during the AIn process, L aims to occupy unsurprising environmental states that minimize the FE (i.e., maximize rewards in reinforcement learning) by learning incrementally novel interactions.

The main contributions of this work can be summarized as follows:

- The dynamic interaction between the expert and a moving object is encoded in a C-GDBN that can be used by L to facilitate the inference and decision-making processes.
- The L's predictive and diagnostic capabilities allow identifying the encountered situation among normal (i.e., L is facing the same situation seen by E) or abnormal (i.e., when L deviates from E's trajectory) and thus guides the exploration-exploitation dilemma. During abnormal situations (i.e., exploration), L learns a set of novel configurations and the associated exploratory actions incrementally, allowing it to come near the reference model and follow the expected trajectory.
- The L's interaction with the environment is determined by a set of actions that minimize the FE measurement, which explains how L expects itself to behave without getting an implicit reward signal from the environment.
- The proposed approach is validated on a real dataset of sensory information collected from two autonomous vehicles. Results show that the proposed approach outperforms conventional RL methods in different learning aspects.

4.2 Model I - Active Inference integrated with Imitation Learning

Learning from experiences is a fundamental capacity of IAs. ASs rely on sensory information that provides data about the environment and internal situations delivered to their perception systems for learning and inference mechanisms. Self-Aware modules can be learned to enable an agent to understand and interact with the surroundings. This section involves two main phases, the offline learning phase, and the online active learning phase. In the former phase, we first provide a Situation model (SM) encoding the dynamic interaction between E and a dynamic vehicle (O). Consequently, we provide L a First-Person model (FP-M), where we assume that L tries to learn sub-optimal behavior by observing the E demonstration. In the latter phase, we present an Active First-Person model (AFP-M) that L can use to update its knowledge while interacting with another moving vehicle (V) in a continuous dynamic environment. All of the mentioned models (i.e., SM, FP-M, and AFP-M) are GDBN that employ graph-based representation to encode various multi-dimensional random variables and represent causal relationships among them [136]. Due to the hierarchical nature, GDBN can express the temporal relationship between high-level variables (capturing abstract semantic information of the world) and low-level distributions (capturing rough sensory information of the environment) with their respective evolution through time. State variables describing the systems' states at a specific time instant k can be categorized as either hidden variables (discrete or continuous) representing the causes affecting the systems' states evolution or measured variables expressing noisy measurements [69]. Since the network size increases over time, performing inference using the entire network would be intractable for all but trivial time duration.

4.3 Offline Learning Phase

4.3.1 Situation Model

SM consists of a C-GDBN representing the interaction of two dynamic vehicles, an expert (E), and a moving agent (O) (see Fig. 4.1). SM relies on multi-modal perception to learn the dynamic evolution of knowledge sets in which an agent is designed to emulate having conscious knowledge of its state and to project the interactive environment, which aims at learning the mapping of different sensory perceptions into exteroceptive and proprioceptive latent information. The SM observes the multi-modal sensorial information as sub-optimal

experiences to predict the dynamic behaviors of the agent and interaction patterns within the dynamic environment.

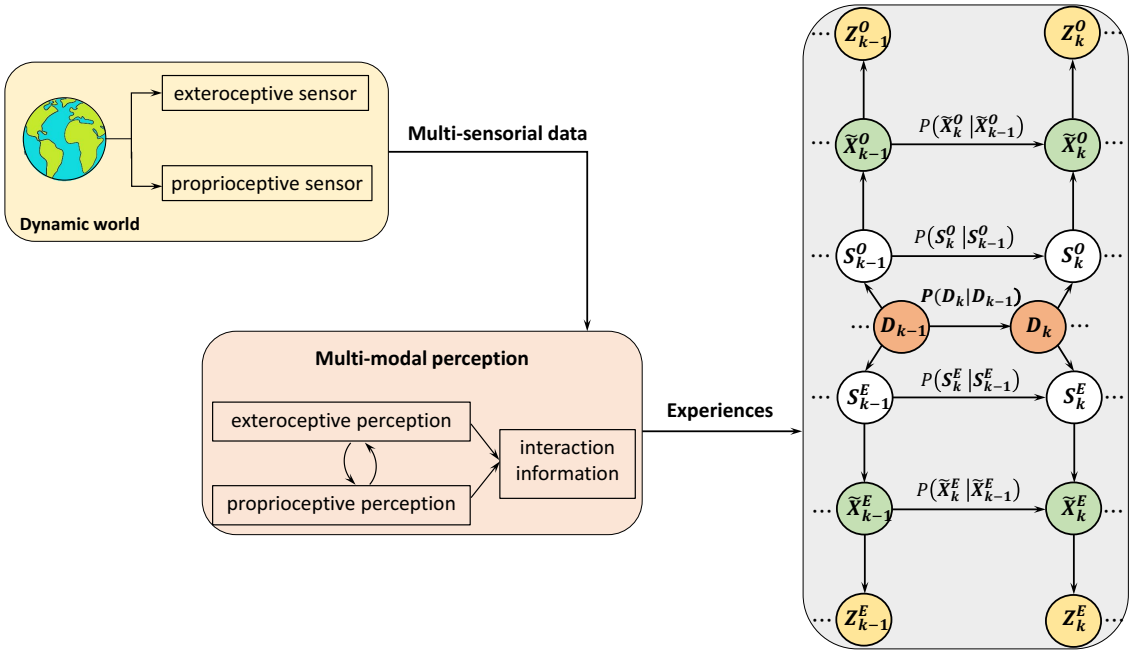


Fig. 4.1 Learning a Situation model.

The agents' observations representing the low level of the hierarchy are described by variables Z_k^i, where $i \in \{\mathrm{E}, \mathrm{O}\}$. At a higher level, the joint hidden continuous Generalized States (GSs) incorporating the dynamics of the two agents at each time instant k can be defined as: $\tilde{\mathrm{X}}_k = [\tilde{\mathrm{X}}_k^{\mathrm{E}} \ \tilde{\mathrm{X}}_k^{\mathrm{O}}]^\intercal$, where $\tilde{\mathrm{X}}_k^{\mathrm{E}}$, $\tilde{\mathrm{X}}_k^{\mathrm{O}}$ denote the GSs of E and O, respectively. The GS related to agent i is defined as a vector composed of the agent's state and its first-order temporal derivative, such that $\tilde{\mathrm{X}}_k^i = [x \ \dot{x}]^\intercal$ where $x \in \mathbb{R}^d$, $\dot{x} \in \mathbb{R}^d$ and d stands for the dimensionality. The observation model describing the relationship between Z_k^i and $\tilde{\mathrm{X}}_k^i$ is defined as:

$$\mathrm{Z}_k^i = \mathrm{H}\tilde{\mathrm{X}}_k^i + v_k, \tag{4.1}$$

where $H = [\mathrm{I}_d \ \ 0_{d,d}]$ is the observation matrix that maps hidden GSs ($\tilde{\mathrm{X}}_k^i$) to measurements (Z_k^i) and v_k is the measurement noise, such that, $v_k \sim \mathcal{N}(0, \mathrm{R})$.

Initially, we assume that the evolution of $\tilde{\mathrm{X}}_k^i$ follows a static equilibrium assumption described by:

$$\tilde{\mathrm{X}}_k^i = \mathrm{A}\tilde{\mathrm{X}}_{k-1}^i + w_k, \tag{4.2}$$

where $\mathrm{A} \in \mathbb{R}^{d \times d}$ is the dynamic matrix and w_k is the process noise, such that $w_k \sim \mathcal{N}(0, \mathrm{Q})$. A Null Force Filter (NFF) [74] is employed to predict $\tilde{\mathrm{X}}_k^i$ according to (4.2). The innovations

encoding the deviations between predictions and observations are calculated by the NFF as:

$$\tilde{\varepsilon}_{\tilde{\mathrm{X}}_t} = \mathrm{H}^{-1}\big(\mathrm{Z}_k^i - \mathrm{H}\tilde{\mathrm{X}}_k^i\big). \tag{4.3}$$

$\tilde{\varepsilon}_{\tilde{\mathrm{X}}_t}$ represents the generalized errors (GEs) that can be clustered in an unsupervised manner using the Growing Neural Gas with utility measurement (GNG-U) [73]. Latter outputs a set $\mathrm{S}^i = \{\mathrm{s}_1^i, \mathrm{s}_2^i, \ldots, \mathrm{s}_{\mathscr{L}_i}^i\}$ of discrete variables (i.e., clusters) representing the so-called vocabulary where $\mathscr{L}_i$ is the total number of clusters.

The joint vocabulary (i.e., $\mathrm{S}^{\mathrm{E}}, \mathrm{S}^{\mathrm{O}}$) describes a specific type of interaction among the two agents at multiple levels (i.e., discrete and continuous levels). Each discrete state represents a region where quasi-linear models are valid to present the interactive dynamical system over time. Since each cluster $\mathrm{s}^i \in \mathrm{S}^i$ is supposed to follow a multivariate Gaussian distribution, it can be represented by its sufficient statistics, specifically, the covariance matrix $\tilde{\Sigma}_{\mathrm{s}_k^i}$ and the generalized mean value $\tilde{\mu}^{\mathrm{s}^i} = [\mu_{Pos}^{s^i}\ \mu_V^{s^i}]$, where $\mu_{Pos}^{s^i}$ and $\mu_V^{s^i}$ represent the mean value of the states (on position) and the mean value of the corresponding derivatives (on velocity), respectively.

An additional vocabulary encoding the dictionary configurations can be defined by $\mathrm{D} = \{\mathrm{D}_1, \mathrm{D}_2, \cdots, \mathrm{D}_{\mathrm{M}}\}$, where $\mathrm{D}_{m,k} = [\mathrm{s}_k^{\mathrm{E}}, \mathrm{s}_k^{\mathrm{O}}]^\intercal$ is an interaction configuration explaining the jointly activated clusters occurring simultaneously in the agents' vocabularies, M is the total number of configurations and $\mathrm{D}_{m,k} \in \mathrm{D}$. Each $\mathrm{D}_{m,k}$ consists of the position and velocity features of the two agents and is defined as:

$$\mathrm{D}_{m,k} = \big[(\mu_{Pos}, \mu_V)^{\mathrm{E}}, (\mu_{Pos}, \mu_V)^{\mathrm{O}}\big]. \tag{4.4}$$

Consequently, the dynamic model defined in (4.2) can be updated as follows:

$$\tilde{\mathrm{X}}_k^i = \mathrm{A}\tilde{\mathrm{X}}_{k-1}^i + \mathrm{B}\mu_V^{s_k^i} + w_k, \tag{4.5}$$

where $\mathrm{B} \in \mathbb{R}^{d \times d}$ is a control model matrix, $\mu_V^{\mathrm{s}_k^i} = [\dot{x}_{s_k^i}, \dot{y}_{s_k^i}]$ is a control vector encoding the agent's velocity (on x and y) associated with s_k^i. The dynamic transitions among the learned configurations at the top level of the hierarchy are encoded in a Transition Matrix (Π) that can be learned by estimating the transition probabilities $\mathrm{P}(\mathrm{D}_{k+1}|\mathrm{D}_k)$ is defined as:

$$\Pi = \begin{bmatrix} \mathrm{P}(\mathrm{D}_1|\mathrm{D}_1), & \mathrm{P}(\mathrm{D}_1|\mathrm{D}_2), & \ldots, & \mathrm{P}(\mathrm{D}_1|\mathrm{D}_{\mathrm{M}}) \\ \mathrm{P}(\mathrm{D}_2|\mathrm{D}_1), & \mathrm{P}(\mathrm{D}_2|\mathrm{D}_2), & \ldots, & \mathrm{P}(\mathrm{D}_2|\mathrm{D}_{\mathrm{M}}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{P}(\mathrm{D}_{\mathrm{M}}|\mathrm{D}_1), & \mathrm{P}(\mathrm{D}_{\mathrm{M}}|\mathrm{D}_2), & \ldots, & \mathrm{P}(\mathrm{D}_{\mathrm{M}}|\mathrm{D}_{\mathrm{M}}) \end{bmatrix}, \tag{4.6}$$

where $\sum_{m}^{\mathrm{M}} \mathrm{P}(\mathrm{D}_p|\mathrm{D}_m) = 1$ such that $p, m \in \mathrm{M}$.

4.3.2 First-Person Model

The First-Person model (FP-M) can be seen as a SM transformed in such a way that allows L to directly use its own observations and generate state series describing its relative state with respect to another interacting moving agent V. It provides L with the capability to imitate the expert motions by generating transformed sequences from SM. A mapping implies defining all GDBN nodes of the created FP-M (DL and CL) and probabilistic dependency models starting from the SM nodes and links. Therefore, FP-M can be considered as an initialization generative switching model represented by a GDBN, which can be used to predict interaction states under the perspective of a learning agent. FP-M depicted in Fig. 4.2 is initialized to allow L to exploit the C-GDBN corresponding to SM that is considered as a pure IL from expert demonstrations.

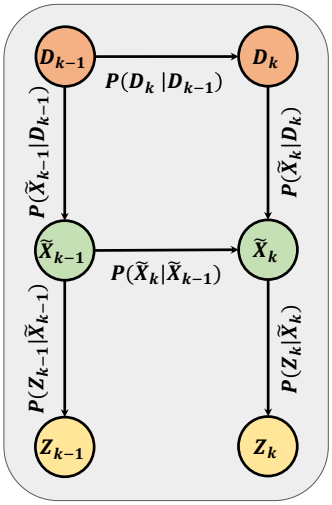


Fig. 4.2 First-Person model.

At the bottom level of hierarchy, the observation (Z_k) of L and V can be mapped onto observations (Z_k^i) of both agents,(E, O) according to the following equation:

$$\mathrm{Z}_k = \left[\mathrm{Z}_k^{\mathrm{E}} - \mathrm{Z}_k^{\mathrm{O}}\right]. \tag{4.7}$$

At the CL (middle level), $\tilde{\mathrm{X}}$ represents the generalized relative distance (consisting of relative distance and relative velocity) between E and O (or between L and V in an ideal IL setting) which are interacting in the environment. The generalized relative distance can be seen as the difference of joint GSs describing the interaction at the CL of the two agents in a specific

configuration (D_m) and defined as:

$$\tilde{\mathrm{X}}_k = \left[\tilde{\mathrm{X}}_k^{\mathrm{E}} - \tilde{\mathrm{X}}_k^{\mathrm{O}}\right] = \left[(x^{\mathrm{O}} - x^{\mathrm{E}}), (\dot{x}^{\mathrm{O}} - \dot{x}^{\mathrm{E}})\right]. \tag{4.8}$$

The relative positions of E and O in SM are illustrated in Fig. 4.3. The relative distance vector is highlighted as the difference in absolute coordinates and velocities of the two objects. The relative distance in FP-M is illustrated in Fig. 4.4 where the relative learner reference system is depicted to highlight the information captured in FP-M.

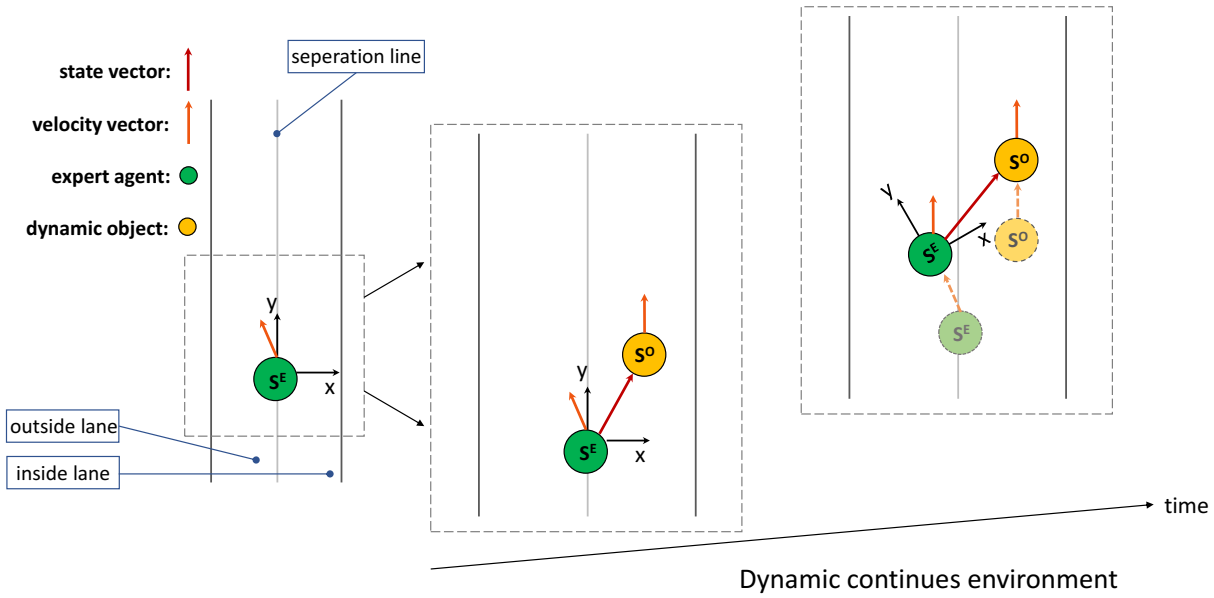


Fig. 4.3 Calculating the relative distance during the agents movements in a non stationary environment.

At the top level of FP-M, the discrete variables represent the learned set of configurations $\mathrm{D}_m \in \mathrm{D}$. In FP-M, L is assumed to take the role of E. Therefore, all clusters related to E should correspond to the clusters describing L states in a certain configuration. By providing a biunivocal mapping between clusters of E and L, the transition probabilities composing Π can characterize the temporal dependencies of discrete series of interactions in FP-M among L and V for the experiences to be imitated. Thus, the transition model can be directly mapped onto FP-M from the corresponding Π of the situation model.

To this end, a configuration $\mathrm{D}_m \in \mathrm{D}$ at the discrete level of FP-M is represented by a joint superstate of each agent at time instant k, i.e., $\mathrm{D}_k = \left[s_k^{\mathrm{L}}, s_k^{\mathrm{O}}\right]$. Thus, the model can predict the expected future configurations based on the dynamic transition rules encoded in the transition

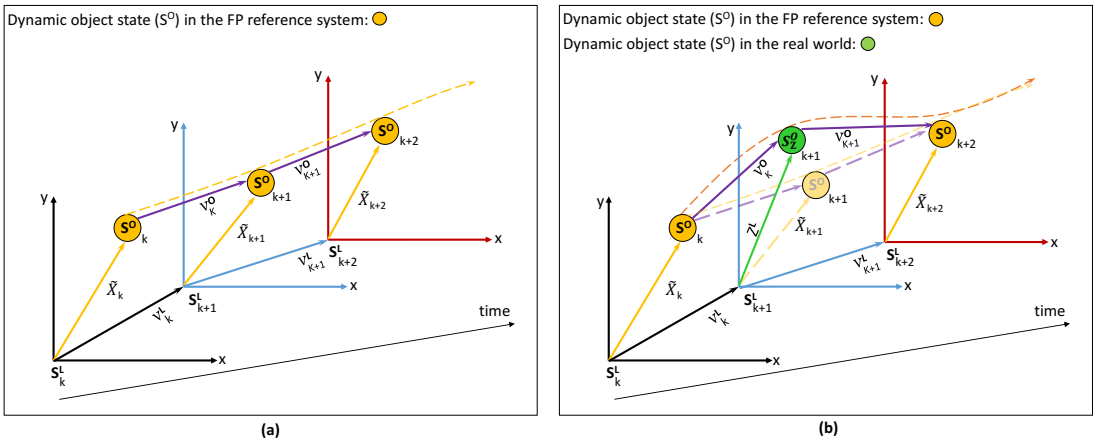


Fig. 4.4 The figure shows the learner agent movements in a continuous dynamic environment by the estimated motion at each time k. The learner state (s^{L}) at each k is the origin of the measurements that the velocity vector (v^{L}) leads it to the next state (s^{L}_{k+1}). (a) shows a normal situation that the learner's interaction with the another dynamic object (s^{O}) is similar with the FP-M's prediction. (b) shows an abnormal situation, where the prediction ($\tilde{\mathrm{X}}$) and the learner observation (Z^{L}) are different due to the different object's velocity (v^{O}) which in-turn brings changes in the behavior of agent.

matrix Π and predict GSs based on the following dynamic model:

$$\tilde{\mathrm{X}}_k = \mathrm{A}\tilde{\mathrm{X}}_{k-1} + B\mu_V^{D_k} + w_k, \tag{4.9}$$

which is characterized by the conditional probability $\mathrm{P}(\tilde{\mathrm{X}}_k|\tilde{\mathrm{X}}_{k-1}, \mathrm{D}_k)$.

4.4 Online Active Learning Phase

During the online learning phase, L utilises a hybrid mechanism combining IL with AIn to describe how it expects itself to behave in a dynamic environment and to learn the best set of actions that it should perform.

4.4.1 Active First-Person model

In this phase, L moves in a dynamic environment and considers its interaction with V in real-time. Therefore, L is endowed with an Active First-Person model (AFP-M) that extends FP-M by adding the active states describing the L's actions in the environment and its influence on the received sensory signals.

L starts with the situation assessment to understand if the current situation has been experienced by the expert agent. This is possible by predicting the relative distance that it

is supposed to observe using FP-M and the actual relative distance measured by L's exteroceptive sensor. When L realizes that current situation is similar to the one encountered by E, then it acts by imitating the expert's behaviour (i.e., selecting the same actions performed by E).

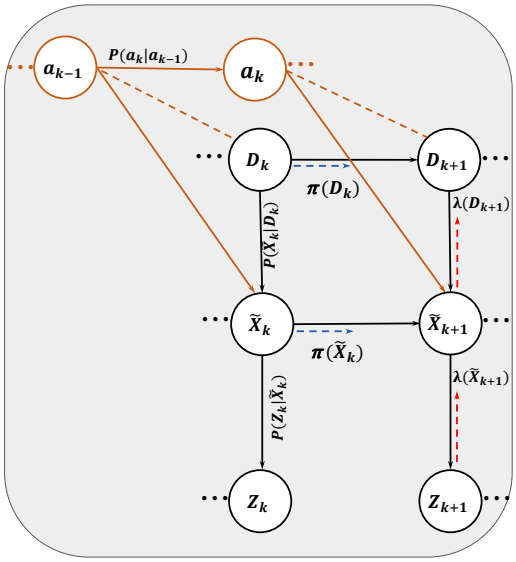


Fig. 4.5 Active First Person model. To run an online learning procedure, the model applies the learner agent's motions (a) to the FP-M at each time instant (orange links).

The active agent (i.e., L), maintains an internal generative model $\mathrm{P}(\mathrm{Z},\tilde{\mathrm{X}},\mathrm{D},a)$ of the prevalent environment expressed in AFP-M (as depicted in Fig. 4.5) and aims to minimize implicitly the difference between what it believes about the environmental states and what it perceives. The AFP-M specifies the joint probability of observations (Z), their hidden causes ($\tilde{\mathrm{X}}$, D) and actions (a). Since the environment is modelled as a Markov Decision Process (MDP), AFP-M can be factorized as:

$$\mathrm{P}(\mathrm{Z},\tilde{\mathrm{X}},\mathrm{D},a) = \mathrm{P}(\mathrm{D}_1)\mathrm{P}(\tilde{\mathrm{X}}_1)\Bigg[\prod_{k=2}^{\mathrm{T}}\mathrm{P}(Z_k|\tilde{\mathrm{X}}_k) \\ \mathrm{P}(\tilde{\mathrm{X}}_k|\tilde{\mathrm{X}}_{k-1},\mathrm{D}_k,a_{k-1})\mathrm{P}(a_{k-1}|\mathrm{D}_{k-1})\Bigg]. \quad (4.10)$$

The proposed AIn approach integrated with IL (**AIL**) involves four main steps: 1) Prediction and Perception, 2) Action selection and 3) FE measurement and 4) Action update. The logic of the AIn approach is reported in **Algorithm 1**.

4.4.2 Prediction and Perception

L employs Particle Filter (PF) to predict the experienced configurations D_k by E and consequently estimates the relative distance $\tilde{\mathrm{X}}_k$ from V at each time step k. At the first iteration ($k = 1$), L relays on prior probability distributions ($\mathrm{P}(\tilde{\mathrm{X}}_1)$, $\mathrm{P}(\mathrm{D}_1)$) to predict the relative distance ($\tilde{\mathrm{X}}_1$) from V and the expected configuration (D_1). In the successive iterations ($k > 1$), L relays on the interactive transition matrix Π to predict future configurations which guides the prediction of the relative distance at the lower level. PF propagates a set of N particles equally weighted using a specific row ($\pi(\mathrm{D}_k)$) in Π as a proposal distribution, such that, $\{\mathrm{D}_{k,n} \sim \pi(\mathrm{D}_k), \mathrm{W}_{k,n} = \frac{1}{N}\}$. For each particle n representing the predicted configuration $\mathrm{D}_{k,n}$, the expected hidden states ($\tilde{\mathrm{X}}^{\mathrm{E}}_{k,n}$, $\tilde{\mathrm{X}}^{\mathrm{O}}_{k,n}$) of E and O can be estimated according to the following dynamic equations:

$$\tilde{\mathrm{X}}^{\mathrm{E}}_{k,n} = \mu^{\mathrm{E}}_{\mathrm{D}_{k,n}} + w_k, \tag{4.11}$$

$$\tilde{\mathrm{X}}^{\mathrm{O}}_{k,n} = \mu^{\mathrm{O}}_{\mathrm{D}_{k,n}} + w_k, \tag{4.12}$$

where $\mu^{\mathrm{E}}_{\mathrm{D}_{k,n}}$, $\mu^{\mathrm{O}}_{\mathrm{D}_{k,n}}$ are associated with clusters $\tilde{\mathrm{S}}^{\mathrm{E}}_{k,n}$ and $\tilde{\mathrm{S}}^{\mathrm{O}}_{k,n}$, respectively, such that $\{\tilde{\mathrm{S}}^{\mathrm{E}}_{k,n}, \tilde{\mathrm{S}}^{\mathrm{O}}_{k,n}\} \in \mathrm{D}_{k,n}$. Then, the relative distance from O can be approximated as follows:

$$\tilde{\mathrm{X}}_{k,n} = \tilde{\mathrm{X}}^{\mathrm{O}}_{k,n} - \tilde{\mathrm{X}}^{\mathrm{E}}_{k,n}. \tag{4.13}$$

Thus, this approximation depends on the hypothesized configuration that explains implicitly the conditional probability $\mathrm{P}(\tilde{\mathrm{X}}_{k,n}|\tilde{\mathrm{X}}_{k-1,n}, \mathrm{D}_{k,n})$. In this sense, L associates itself to a specific configuration ($\mathrm{D}_{k,n}$) and predicts the relative distance from the current dynamic object V which it is dealing with. L receives observations (Z_k) through its exteroceptive sensor and realize actions to be done by its actuators. Once a new Z_k is given - describing the relative distance between the L and V - L can evaluate if the situation it is experiencing has already faced by E in order to make a decision on selecting an action (i.e., the decision between exploitation and exploration).

Diagnostic messages ($\lambda(\tilde{\mathrm{X}}_k)$ and $\lambda(\mathrm{D}_k)$) propagated from the bottom level towards higher levels inside AFP-M allows defining an abnormality measurement to evaluate how much current observation supports predictions as well as updating the belief in hidden variables. The model computes the anomaly (Ω) by measuring the cosine similarity $(\cos(\theta))$ between the observed relative distance ($\tilde{\mathrm{Z}}_k = d^L_z$) and the predicted relative distance ($\tilde{\mathrm{X}}_{k,n}$) associated with each propagated particle as follows:

$$\Omega_{k,n} = \cos(\theta) = \frac{\tilde{\mathrm{Z}}_k \,.\, \tilde{\mathrm{X}}_{k,n}}{||\tilde{\mathrm{Z}}_k||\ ||\tilde{\mathrm{X}}_{k,n}||}. \tag{4.14}$$

The lower the angle θ, the lower the abnormality value, so more similarity is achieved. Particles gain weight according to their similarity with the observation. A high similarity value (the lower angle) gains more weight $(W_{k,n})$ than particles with low similarity. Message $\lambda(\mathrm{D}_k)$ is used to update particles' weights and it is defined as:

$$\lambda(\mathrm{D}_k) = \lambda(\tilde{\mathrm{X}}_k)\mathrm{P}(\tilde{\mathrm{X}}_k|\mathrm{D}_k), \tag{4.15}$$

where $\lambda(\tilde{\mathrm{X}}_k) = \mathrm{P}(\mathrm{Z}_k|\tilde{\mathrm{X}}_{k,n})$ is a multivariate Gaussian distribution such that $\lambda(\tilde{\mathrm{X}}_k) \sim \mathcal{N}(\mathrm{Z}_k, v_k)$ and $\lambda(\mathrm{D}_k)$ is a discrete probability distribution. Consequently, particles' weights can be updated as follows:

$$W_{k,n} = W_{k,n} \times \lambda(\mathrm{D}_k). \tag{4.16}$$

4.4.3 Action Selection

The updated particles' weights allow L to decide whether to exploit actions by imitating the E's behaviour or to explore new actions that may yield lower FEs (higher rewards) in the future. The decision between exploration and exploitation is based on two parameters, namely, the exploration rate (ε) and a varying threshold (t). The former is defined as:

$$\varepsilon_k = 1 - \alpha_k, \tag{4.17}$$

where α is the largest weight among all the N particles measuring the likelihood between the current L configuration and the reference configuration, such that:

$$\alpha_k = \max_n W_{k,n}, \tag{4.18}$$

where $0 \leq \alpha \leq 1$. So, if α_k is near 1, ε_k becomes very low which means that current observation matches L's expectation and so it can exploit the same actions performed by E. However, in other cases it might appear that α is not too high (e.g., below 0.5). In this case, it is required to evaluate the anomaly level associated with the particle index that has the maximum weight and the defined t based on a trial-and-error process. Thus, action generation process depends on the decision made by L whether to explore or exploit and it is defined as:

$$a_k \sim \begin{cases} \mu_V^{\mathrm{D}_k^\beta} = \arg\max_{a_k} \mathrm{Q}(\mathscr{A}, \mathrm{D}_k^\beta), \text{ if } \varepsilon < t \text{ (exploitation)}, \\ \textit{random} \text{ from } \mathscr{A}^+, \text{ if } \varepsilon \geq t \text{ (exploration)}, \end{cases} \tag{4.19}$$

where a_k are the active states (i.e., actions) realizing the top level of AFP-M, $\mathscr{A} = \{\mathscr{A}^{\mathrm{E}}, \mathscr{A}^{+}\}$, where $\mathscr{A}^{\mathrm{E}} = \{a_1^{\mathrm{E}}, a_2^{\mathrm{E}}, \ldots, a_{\mathrm{Y}}^{\mathrm{E}}\}$ is a set of actions performed by E and encoded in SM that L aims to imitate during exploitation and $\mathscr{A}^{+} = \{a_1, a_2, \ldots, a_8\}$ is a set of predefined actions realizing 8 different directions[1] used during exploration. In addition, D_k^{β} is the most similar reference configuration to the observed one and β is the particle's index with the maximum weight associated with (4.32) defined as:

$$\beta = \arg\max_{n} \ (W_{k,n}), \tag{4.20}$$

Moreover, during exploration, L saves the novel configurations D_k^{+} (not seen by E) that is experiencing along with the performed actions $a_k^{+} \in \mathscr{A}^{+}$ in a set ($\mathscr{C}$). After finishing a certain experience L clusters all the pairs $[\mathrm{D}_k^{+}, a_k^{+}]$ saved in $\mathscr{C}$ by employing the GNG. The latter outputs a set of clusters representing the new configurations (D^{++}) that can be appended incrementally to the probabilistic Q-table (Q) that is defined as:

$$\mathrm{Q} = \begin{bmatrix} \mathrm{P}(a_1^{\mathrm{E}}|\mathrm{D}_1) & \ldots & \mathrm{P}(a_1^{\mathrm{E}}|\mathrm{D}_{\mathrm{M}}) & \mathrm{P}(a_1^{\mathrm{E}}|\mathrm{D}^{++}) & \ldots \\ \mathrm{P}(a_2^{\mathrm{E}}|\mathrm{D}_1) & \ldots & \mathrm{P}(a_2^{\mathrm{E}}|\mathrm{D}_{\mathrm{M}}) & \mathrm{P}(a_2^{\mathrm{E}}|\mathrm{D}^{++}) & \ldots \\ \vdots & \ddots & \vdots & \vdots & \\ \mathrm{P}(a_{\mathrm{Y}}^{\mathrm{E}}|\mathrm{D}_1) & \ldots & \mathrm{P}(a_{\mathrm{Y}}^{\mathrm{E}}|\mathrm{D}_{\mathrm{M}}) & \mathrm{P}(a_{\mathrm{Y}}^{\mathrm{E}}|\mathrm{D}^{++}) & \ldots \\ \mathrm{P}(a^{++}|\mathrm{D}_1) & \ldots & \mathrm{P}(a^{++}|\mathrm{D}_{\mathrm{M}}) & \mathrm{P}(a^{++}|\mathrm{D}^{++}) & \ldots \\ \vdots & & \vdots & \vdots & \end{bmatrix}, \tag{4.21}$$

where $\sum_y^{\mathrm{Y}} \mathrm{P}(a_y^{\mathrm{E}}|\mathrm{D}_m) + \sum_{e=1} \mathrm{P}(a_e^{++}|\mathrm{D}_m) = 1$ and $\sum_y^{\mathrm{Y}} \mathrm{P}(a_y^{\mathrm{E}}|\mathrm{D}^{++}) + \sum_{e=1} \mathrm{P}(a_e|\mathrm{D}^{++}) = 1$ such that $m \in \mathrm{M}$ and $y \in \mathrm{Y}$, $a^{++} = \mu_V^{\mathrm{D}^{++}}$ are the new explored actions that can be exploited in the future. In addition, L updates the transition model defined in (4.6) by adding new rows and columns which are related to the new configurations incrementally.

In the *exploitation* phase, if the current configuration is an observed one by E, the learning agent takes the adapted expert action from prior knowledge by activating the most similar reference configuration ($\dot{\mathrm{D}}_k = \mathrm{D}_k^{\beta}$) to the current L configuration at the real-time and consequently select the suitable action (i.e., representing the L's motion) according to $\mathrm{P}(a_k|\mathrm{D}_k^{\beta})$ encoded in Q. After that, by adapting the expected motion $\mathrm{P}(\tilde{\mathrm{X}}_k|\mathrm{D}_k)$ at time k through the active states $\mathrm{P}(a_k|\mathrm{D}_k, \tilde{\mathrm{X}}_k)$, the L agent transits to a new configuration realized by $\mathrm{P}(\mathrm{D}_{k+1}|a_k, \mathrm{D}_k)$. Thus, the conditional prior $\mathrm{P}(a_k|\mathrm{D}_k, \tilde{\mathrm{X}}_k)$ is maximized after having been initialized according to demonstration to select the best action a_k given the current configuration and state.

[1]The 8 directions are North, South, East, West, North-West, North-East, South-East, South-West.

Besides, in *exploration* phase, if the mismatch between predictions and observation is too high, the model can not apply the direct imitation of E by taking a learned action from SM. However, if L faces an anomaly, the learning model considers it as an unseen situation. Hence, the newly explored configurations are added to the reference configurations (incremental learning model). Moreover, the model corresponds a set of possible actions with equal selection probabilities to the newly added configuration that L can take randomly to move in the environment. The selection probabilities are modified through the online learning phase. The presented learning procedure aims at converging at some optimal policy to the lower probability of taking a random action over time as the agent becomes more confident with its estimations. During exploration, L aims to take the best set of actions that can approach it to the reference configurations (i.e., reference vocabulary realizing the expert's behaviour in dealing with a dynamic object in the environment).

AFP-M improves the L's behavior during the training by minimizing the divergence between SM and the AFP-M to decrease the loss cost of imitation, besides dealing with the abnormalities to decline the collision probability or going out of boundaries. During the active learning phase, the selection probabilities related to each movement are recorded by $\mathrm{P}(a_k|\mathrm{D}_k)$ and updated at each time instance. L needs to exploit the expert demonstrations to minimize the global FE by modifying the transition policies. It also needs to explore through the new experiences to make better action selections in the future. L must modify its actions several times to gain a reliable prediction with a low imitation loss cost and FE measurement on a stochastic task while switching between exploration and exploitation dynamically.

4.4.4 Free Energy Measurement

The IA faces multiple tasks that need ample action space. Therefore it is challenging to acquire an appropriate policy by a onefold reward strategy. We aims finding a reward function R that could explain the expert policy from demonstrations. The proposed approach endows the IA with the capability of estimating the imitation cost (i.e., reward) in terms of FE at multiple levels. Minimizing the FE (i.e., maximizing rewards in RL context) ensures a dynamic equilibrium between the L and its prevalent environment.

The FE measurements are based on the AFP-M hierarchy's messages (messages passing from top-to-down and bottom-to-up). The message (λ) passing from lower nodes to upper nodes has a diagnostic ability used to adjust the expectations (predictions by inter-slice links π) given a sequence of observations. Comparing predictive and diagnostic messages allows the detection whether new observations are similar to previously learned situations encoded in FP-M. Suppose predictions from FP-M are not compliant with observations, then the

Algorithm 1 Active Inference integrated with IL (AIL)

Input: Π, $\mathbf{D} = \{\tilde{\mu}^{\mathbf{O}} - \tilde{\mu}^{\mathbf{E}}\}$, Q $\leftarrow$ Transition Matrix, Configurations, QTable

for $k = 1$ to K $\leftarrow$ Time evolution **do**
  for $n = 1$ to N $\leftarrow$ Particles **do**
    Prediction at the discrete level:
    if $k == 1$ $\leftarrow$ Initial iteration **then**
      $\mathrm{P}(\tilde{\mathrm{X}}_1) \sim \mathscr{N}(\mu_{\tilde{\mathrm{X}}_1}, \Sigma_{\tilde{\mathrm{X}}_1}) \leftarrow$ prior distribution
      Sample $\tilde{\mathrm{X}}_k^{(n)} \sim \mathrm{P}(\tilde{\mathrm{X}}_1)$
      $\mathrm{P}(\mathrm{D}_1) = \mathscr{U}\{1, |\mathrm{D}_m|\} \leftarrow$ uniform distribution
      Sample $\mathrm{D}_{k,n} \sim \mathrm{P}(\mathrm{D}_0)$
      $W_{k,n} = \frac{1}{N} \leftarrow$ particle weight
    else if $k > 1$ **then**
      $\mathrm{D}_{k,n} \sim \mathrm{TM}(\mathrm{D}_{k-1,n}) \leftarrow$ proposal from transition matrix
      Prediction at the continuous level:
      $\tilde{\mathrm{X}}_{k,n} = \mu^{\mathrm{E}}_{\mathrm{D}_{k,n}} + w_k \leftarrow$ Expert's Mean of cluster $\mathrm{S}^{\mathrm{E}}_{k,n}$
      $\tilde{X}^{O}_{k,n} = \mu^{O}_{D_{k,n}} + w_k \leftarrow$ Object's Mean of cluster $S^{O}_{k,n}$
      $d_{k,n} = \tilde{X}^{O}_{k,n} - \tilde{X}_{k,n} \leftarrow d_{k,n} \in \mathbb{R}^{1,4} \leftarrow$ distance vector
      $\pi(\tilde{X}_{k,n}) = d_{k,n} \leftarrow$ Predictive msg
      $\pi(\tilde{X}_{k,n}) \sim \mathscr{N}(\mu_{d_{k,n}}, \Sigma_{d_{k,n}}) \leftarrow$ Predictive msg
    end if
    Receiving the learner observation Z_k
    $\lambda(\tilde{X}_{k,n}) = p(Z_k|\tilde{X}_{k,n})$
    $\lambda(D_k) = D_B\big(\lambda(\tilde{X}_k), p(\tilde{X}_k|D_k)\big) \leftarrow$ unique for all particles
    Anomaly indicator:
    $\Omega = D_\theta\big(\pi(\tilde{X}_{k,n}), \lambda(\tilde{X}_{k,n})\big)$
    Update:
    $W_{k,n} = W_{k,n} \times \lambda(D_k) \leftarrow$ updated weight
    RIS resampling
    $W_{k+1,n} = \frac{1}{N}$
  end for
  Action selection:
  $\beta = \arg\max_n \; W_{k,n}$
  $\alpha = \max_n \; W_{k,n}$
  The corresponded configuration to β presents the activated reference configuration ($\dot{\mathrm{D}}_k$)
  $\dot{\mathrm{D}}_k = D^{\beta}_k$.
  $\rho \leftarrow$ Threshold for adding new configuration
  $\varepsilon = 1 - \alpha \leftarrow$ exploration rate
  if $\varepsilon < \rho$ **then**
    $a_k \sim \mu_V^{D^{\beta}_k} = \arg\max_{a_k} \; \mathrm{Q}(\mathscr{A}, \mathrm{D}^{\beta}_k) \leftarrow$ exploitation
  else
    $a^{+}_k \sim$ random from $\mathscr{A}^{+} \leftarrow$ exploration
    save $[\mathrm{D}^{+}, a^{+}_k]$ in $\mathscr{C}$
  end if
  $\mathrm{Q}^{*} =$ `FREE ENERGY MEASUREMENT`$\big(\mathrm{Q}, a_k, \hat{\mathrm{D}}_k\big)$
end for

model considers the current experience as an anomalous experience, and so it should be adapted to by learning new situations and generating new semantic information.

The diagnostic messages evaluate the distinction between the expectation and evidence at two abstraction levels. We theoretically extend the FE measurement by estimating both the prior and posterior policy at continuous and discrete levels.

The goal is to allow L to maximize the likelihood by using the FE as a control metric. Under the FE principle, L uses the likelihood estimation of the prior hidden states based on the active reference configuration ($\dot{D}$) and the observations. The determined prior by the hidden states and actions at the previous time instant can change the L's future policy.

FE measurement at the continuous level (F_{CL})**.** AFP-M allows evaluating how much the sensory measurements support predictions and thus evaluating if the selected actions were good or wrong by relying on the FE. The FE at the CL can be computed by evaluating the distinction between the predictive message $\pi(\tilde{\mathrm{X}}_k)$ and the diagnostic message $\lambda(\tilde{\mathrm{X}}_k)$ after taking action a_{k-1} under both exploration and exploitation condition.

Thus, the performed action by the learning agent (a^L_{k-1}) guides the system to calculate the expected FE [54] at the continuous level (F_{CL}) based on the Kullback Leibler-Divergence ($\mathscr{D}_{\mathscr{KL}}$) [83] between $\pi(\tilde{\mathrm{X}}_k)$ and $\lambda(\tilde{\mathrm{X}}_k)$. Hence, the expected FE can be expressed as:

$$F_{CL} = \mathscr{D}_{\mathscr{KL}}\left(\lambda(\tilde{\mathrm{X}}_k)||\pi(\tilde{\mathrm{X}}_k)\right) = \int \lambda(\tilde{\mathrm{X}}_k)\log\left(\frac{\lambda(\tilde{\mathrm{X}}_k)}{\pi(\tilde{\mathrm{X}}_k)}\right)\mathrm{d}\tilde{\mathrm{X}}_k. \tag{4.22}$$

Our goal is to find a policy such that the L's behavior matches the reference demonstrations. For this purpose, our objective is to minimize the divergence between what L is expecting to observe after taking a certain action and what it is really observing. L believes that a certain action allows it to imitate correctly the E's behavior during exploitation or allows it to approach towards the E's reference vocabulary as soon as possible during exploration.

FE measurement at the discrete level (F_{DL})**.** The FE measurement at the DL (F_{DL}) is computed by employing the Mahalanobis distance ($\mathscr{D}_{\mathscr{M}}$) [38] to calculate the distinction between the action selected by L (a^{L}_k) and the E's estimated action (a^{E}_k) from the activated reference configuration $\mu^{\dot{D}}_V$, defined as:

$$F_{DL} = \mathscr{D}_{\mathscr{M}}\left(a^L_k, a^{\mathrm{E}}_k\right), \tag{4.23}$$

where $a^{\mathrm{E}}_k = max\ \mathrm{Q}(:,\dot{\mathrm{D}}_k)$.

Global FE ($\mathscr{G}$)**.** If the L agent is in a observed configuration (exploitation case) $\mathscr{G}$ is defined as:

$$\mathscr{G} = F_{CL}, \tag{4.24}$$

Algorithm 2 Free Energy measurement

function FREE ENERGY MEASUREMENT(Q, a_k^L, $\dot{\mathrm{D}}_k$)
  $a_k^E \sim max\ \mathrm{Q}(:, \dot{\mathrm{D}}_k)$
  $S_k^E \sim \dot{\mathrm{D}}_k \leftarrow$ Activated expert cluster
  $\mu_{V,k}^E, \tilde{\Sigma}_k^E \sim S_k^E \leftarrow$ Mean velocity and covariance of the expert cluster
  $V^L \sim a_k^L \leftarrow$ velocity vector of learner
  $\hat{v}_k \sim (V_k^L - \mu_{V,k}^E)$
  $F_{DL} = \mathscr{D}_{\mathscr{M}}\left(a_k^L, a_k^{\mathrm{E}}\right) = \sqrt{\hat{v}_k^T (\Sigma_k^E)^{-1} \hat{v}_k} \leftarrow$ The FE at current time (4.23)
  Calculate F_{CL} using (4.22) $\leftarrow$ The excepted future FE
  if the L is exploring: **then**
    $\mathscr{G} = \mathbb{E}(F_{CL}, F_{DL}) \leftarrow$ Global Free Energy
  else
    $\mathscr{G} = F_{CL} \leftarrow$ Global Free Energy
  end if
  Update Q^* using (4.26)
  return Q^*
end function

Otherwise, if it experiences a new configuration or improves the action selection regarding the recorded explored states, $\mathscr{G}$ be expressed as:

$$\mathscr{G} = \mathbb{E}(F_{CL}, F_{DL}), \tag{4.25}$$

To sum up, by improving the action selection to minimize F_{DL} at each time instant, L is able to have a more similar prediction to the future expected observation after taking action, which causes decreasing F_{CL} as well. In the end, the model is able to decline the global FE through the message passing between the multi-levels.

4.4.5 Action Update

AFP-M takes advantage of both discrete and continuous levels dynamically to decrease global loss by improving the action selection through the online learning procedure. Our objective is to minimize the long-term cost by taking down the global FE measurements defined in (4.24) and (4.25). L adapts the action selection process by updating the Q-table defined in (4.21) based on the global FE. Since the Q table used in this work is a probabilistic table, it can be written in a probabilistic form as

$$\mathrm{Q}^* = (1 - \eta)\mathrm{P}(a_{k-1}|\mathrm{D}_{k-1}) + \eta\left[(1 - \mathscr{G}) + \gamma \max_{a_k} \mathrm{P}(a_k|D_k)\right]. \tag{4.26}$$

where η is the learning rate that controls how quickly the learning agent adopts to the explorations imposed by the environment, $\mathscr{G}$ is the normalized global FE measurement with a range from 0 to 1, and γ is a discount factor as in the general case of RL algorithms.

4.5 Simulation and Performance Evaluation - Model I

4.5.1 Experimental Data Set

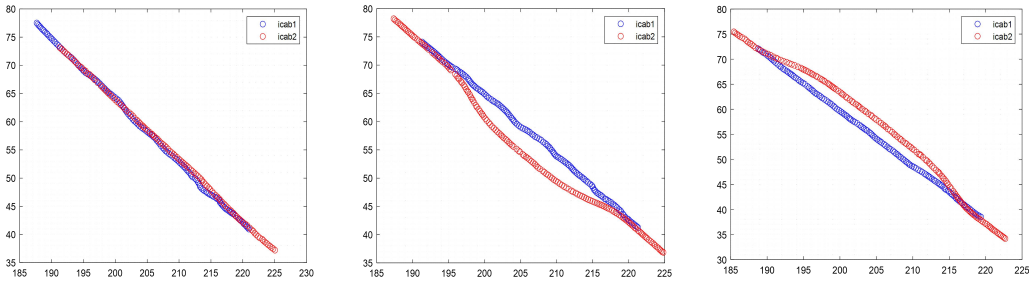
The proposed framework is validated using a real dataset consisting of multisensorial information collected from two autonomous vehicles, 'iCab 1' and 'iCab 2' [91] (see Fig. 4.6 and Fig. 4.7). The vehicles positional information and the corresponding velocities are obtained from the odometry module. This work considers three scenarios:

Fig. 4.6 Autonomous vehicles: iCab 1 and iCab 2.

Fig. 4.7 the yellow parts shows the experimental zone.

- **lane-keeping scenario (following behavior):** iCab 2 follows another agent (iCab 1) as shown in sub-Fig. 4.8-(a) and aims to keep a safe distance from iCab 1. The latter plays the role of a dynamic obstacle in the environment with a higher speed than iCab 2.
- **lane-changing scenario (overtaking behavior) - left side:** iCab 2 overtakes iCab 1 (considered as a dynamic obstacle) to change the lane without collision. In this scenario, iCab 2 has a higher speed than iCab 1, where iCab 2 overtakes from the left side of iCab 1 as depicted in sub-Fig. 4.8-(b).
- **lane-changing scenario (overtaking behavior) - right side:** In this scenario iCab 2 has a higher speed than iCab 1 where iCab 2 overtakes from the right side of iCab 1 as shown in sub-Fig. 4.8-(c).

Sensory data representing positional information from these experiments are used to learn the dynamic interaction between iCab 1 (which plays the role of a dynamic object, i.e., O) and iCab 2 (which plays the role of an expert, i.e., E) encoded in the SM that L will use to imitate the E's demonstrations.



(a) (b) (c)

Fig. 4.8 icab interactions. In (a) iCab 2 follows iCab 1, (b) shows iCab 2 overtakes iCab 1 from the left side, and in (c), iCab 2 overtakes iCab 1 from the right side.

4.5.2 Offline Learning Phase

This section shows the process of learning the SM from collected data during different scenarios. The NFF is used as an initial filter employed on the data collected in the lane-keeping and lane-changing scenarios. NFF outputs the GEs defined in (4.3), which can be clustered using GNG that outputs a set of discrete clusters representing the discrete regions of the trajectories generated by E and O. The joint clusters define the set of configurations (defined in (4.4)) that encode the dynamic interaction among the two agents. Fig. 4.9-(a)-(b)-(c) illustrates the generated clusters in different scenarios and Fig. 4.9-(d)-(e)-(f) shows the corresponding transition matrices.

4.5.3 Online Learning Phase

During the online active learning phase, the AFP-M relies on the FP-M, which has been initialized using the situation model. Thus, the DL in the three models represents the learned configurations during the offline phase. The total number of configurations is 60, then the initial Q-table contains 60 configurations and it is defined as follows:

$$\mathrm{Q} = \begin{array}{c} \\ a_1^{\mathrm{E}} \\ a_2^{\mathrm{E}} \\ \vdots \\ a_{60}^{\mathrm{E}} \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{D}_1 & \mathrm{D}_2 & \dots & \mathrm{D}_{60} \end{array} \\ \begin{bmatrix} \frac{1}{60} & \frac{1}{60} & \cdots & \frac{1}{60} \\ \frac{1}{60} & \frac{1}{60} & \cdots & \frac{1}{60} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{60} & \frac{1}{60} & \cdots & \frac{1}{60} \end{bmatrix} \end{array} \tag{4.27}$$

The experiments are done in a simulated environment. For having a fair comparative evaluation, all the experiments are considered with fixed steps. The algorithm is run over

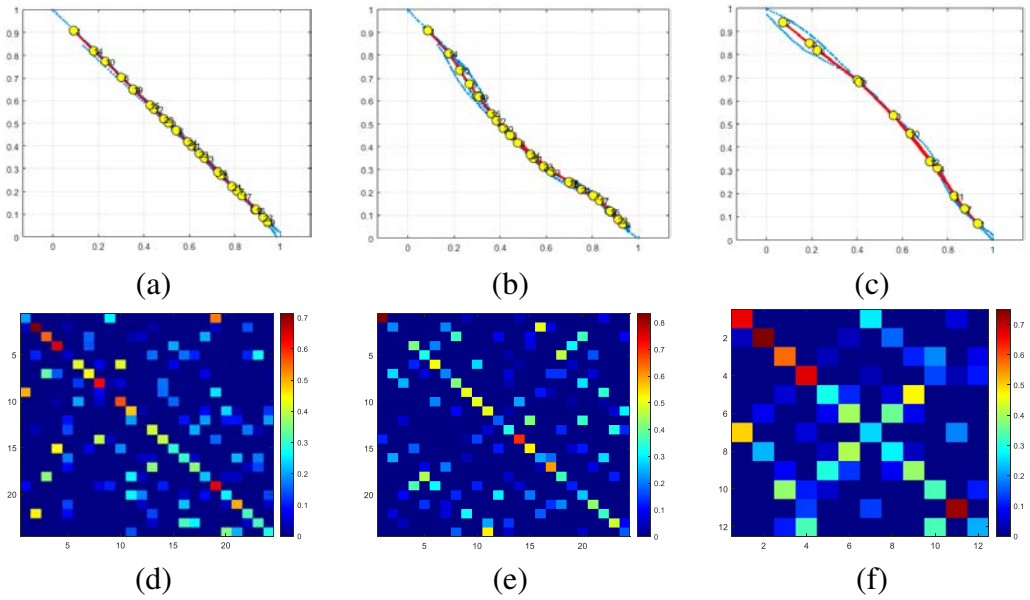


Fig. 4.9 Learning the situation model. a) Clustering of GEs in the lane-keeping scenarios, b) Clustering of GEs in the lane-changing from the left side scenario, and c) Clustering of GEs in the lane-changing from the right side scenario. Sub-figures (d), (e), and (f) are the corresponding transition matrices to sub-figures (a), (b), and (c), respectively.

500 episodes from different start positions to train the learner L. For each iteration during an episode, L is trained to learn how to behave with another moving agent V in a dynamic environment. Each episode consists of 10 iterations, i.e., L tries 5k iterations by 500 different start positions to learn the policies.

Moreover, the FE measurement at the CL helps the learner determine a safe distance from the moving object. The safe distance allows the learner agent to continue lane-keeping without collision probability. At each time instant k, the AS finds the minimum and the mean value of F_{CL}, which is calculated by the KL divergence defined in (4.22). After that, by calculating the differential of the corresponding distance vector's length to the values ($|\Delta d_i|$), the measured safe distance determines a threshold for L, which is changed dynamically at each time instant during the online learning phase until the completion of the trial. The dynamic model uses the safe distance to record the estimations in two Q-tables for the safe zone and the warning zone. In the safe zone, the higher transition probability relates to lane-keeping. On the other side, in the warning zone, the higher transition probability leads the agent to lane-changing to decline the collision probability. The estimations are separated based on L situation during the online learning phase to facilitate and accelerate the making decision during exploiting the learned tables.

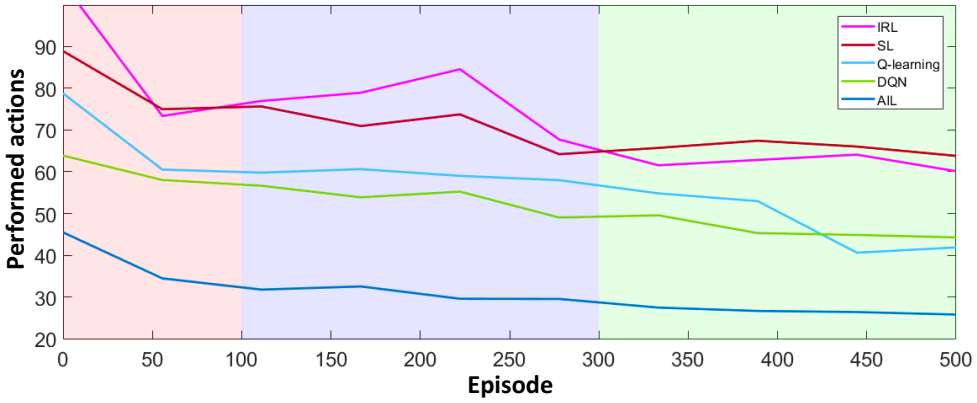


Fig. 4.10 The number of performed actions during the learning phase.

The performance of the proposed method is evaluated in different experiments and compared it with four learning algorithms, namely, the general value-based Q-learning, double Q-Network, IRL (when an optimal expert is available), and self-learning in RL context (when optimal expert data is not available). Performance evaluation considers two main issues, action selection, and imitation loss.

4.5.4 Action Selection

L predicts the configurations (D_m) visited by E by employing PF and then estimates the relative distance from V to decide whether to imitate the E's actions (i.e., exploitation) or to explore new actions. Initially, PF propagates $N = 10$ particles equally weighted ($W = \frac{1}{N} = \frac{1}{10}$) by relaying on the Π (at the first time instant $k = 1$, PF generate samples from a uniform distribution). Action selection realizes an essential process to reach the goal targeted by the agent (e.g., following or overtaking the other agent). The number of performed actions describes the effort made by the agent to accomplish a task. A good policy requires fewer actions and less time to reach the goal, while a lousy policy requires more actions and time.

Fig. 4.10 shows the mean of actions taken by L for each episode during the online learning phase using different methods. From the figure, we can observe that L adopting the proposed approach (AIL) performs fewer actions compared to other methods. This can be explained by the fact that initializing the FP-M using the SM can decrease the exploration rate. Moreover, exploiting sub-optimal expert demonstrations at similar states plays a vital role in driving in a shorter time than exploring the environment from scratch. The threshold ρ has a great impact on the exploration rate, we train the L agent 11 times with different ρ values in the range $[0, 1]$. By considering the success rate obtained by each ρ value, we pick the best ρ value providing the maximum success rate as shown in Fig. 4.11.

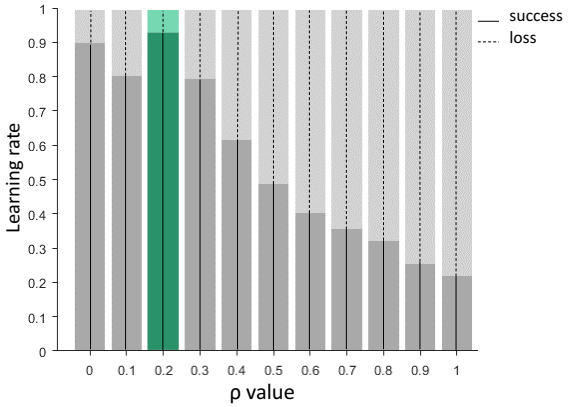


Fig. 4.11 ρ is a threshold that plays the control role to separate the exploration and exploitation mode. We trained the model with different ρ values to find the most suitable one by trial and error. The green bar is the selected one.

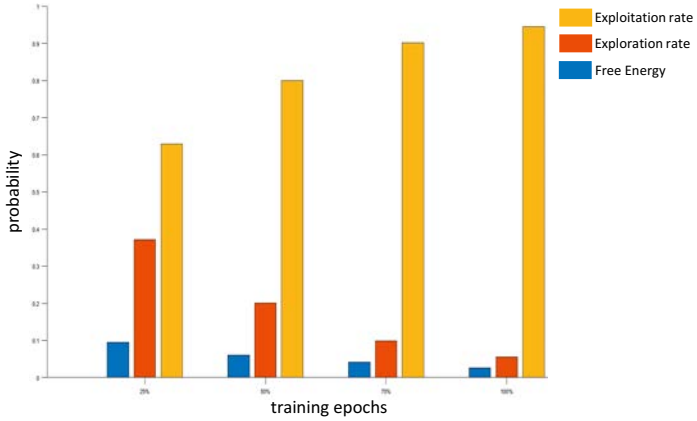


Fig. 4.12 Illustration of the exploitation and exploration rates after each training quarter and their effect on the FE measurement.

In addition, updating particles' weights to adjust the action selection procedure allows L to avoid abnormalities and adapt to new experiences. Fig. 4.12 demonstrates how the exploitation and exploration rates affect the FE measurement during the learning phase. Refining the action selection can adapt to new experiences and minimize the FE measurements.

Balancing exploration and exploitation is one of the most challenging tasks in RL. The imbalance between exploration and exploitation might adversely affect learning performance. On the one hand, the domination of exploration would obstruct the agent to maximize short-term reward, i.e., exploratory actions could lead an agent to collect a higher negative reward in the short run. On the other hand, if a learning approach is dominated by exploitation, an agent performs actions that could get it stuck in local minima or suboptimal solutions.

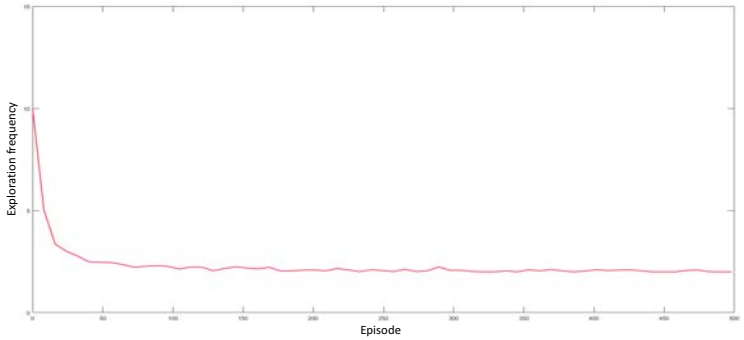


Fig. 4.13 Exploration frequency. It shows after how many explored actions the learner goes back to the exploitation mode (the average number for each episode).

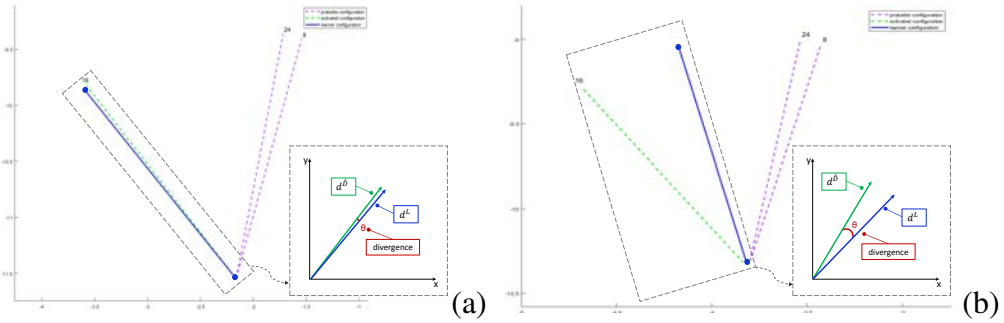
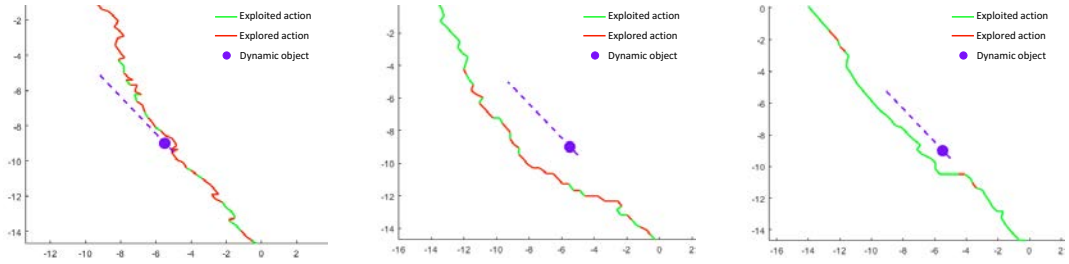


Fig. 4.14 This example illustrates when the learner agent is in the exploitation case (a) or exploration case (b). Purple lines show the relative distance from the most probable configurations, while the green line represents the relative distance from the activated configuration. The learner exploits the activated configuration, leading to a lower divergence, and during the exploration, The learner takes an exploratory action because the divergence between the learner configuration and the activated one is more than ρ. (θ between blue and green vectors).

Fig. 4.13 shows the frequency of the exploratory actions, L is trained to have an equal opportunity to gain new knowledge from the environment's dynamics and follow the expert demonstrations to accomplish its mission (see Fig. 4.14-(a) - (b)).

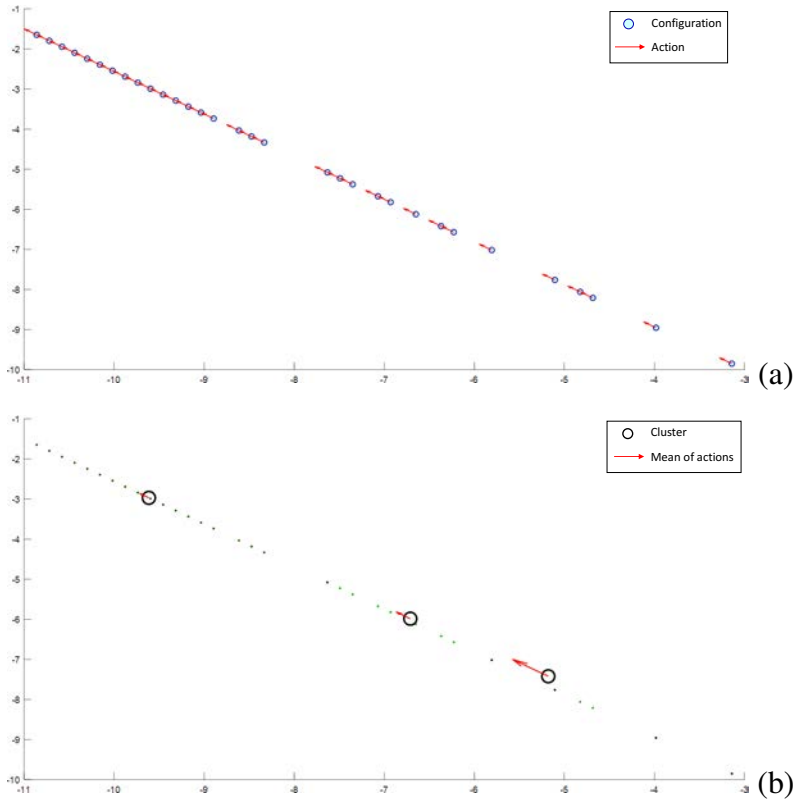
Improving the action selection skill leads L to perform more successful movements in the dynamic environment, as shown in Fig. 4.15. When L enters the exploration stage in a certain episode, it saves all the newly explored configurations along with the performed actions. Then, L clusters those saved pairs (i.e., new configurations and actions) as discussed in Section 4.4.3. The newly explored configurations and actions are clustered for two reasons: to calculate the mean action value of the corresponding clusters in order to have comparable data with the FP model and to avoid recording too many configurations in the Q-table.

In each step, the newly learned clusters are appended incrementally to the model, Fig. 4.16 and Fig. 4.17 describe the clustering process of the new configurations in two scenarios



(a) (b) (c)

Fig. 4.15 This figure shows three trajectories in different time slots of online learning. In (a), the learner experiences new actions by exploration. (b) shows by balancing the exploration and exploitation, the learner improves the action selection, and (c) demonstrates the learner can decrease the explored action and make suitable decisions concerning the dynamic object.



(a) (b)

Fig. 4.16 Clustering the explored configurations in the lane-keeping scenario. (a) shows all-new exploration by the learner through one step and (b) shows the clustered configurations and the corresponding mean action value to them.

related to lane-keeping and lane-changing. New experiences are modified by the action selection by exploiting new appended actions through the online learning phase and resolving L's uncertainty about the surrounding environment. Fig. 4.18 and Fig. 4.19 illustrate the

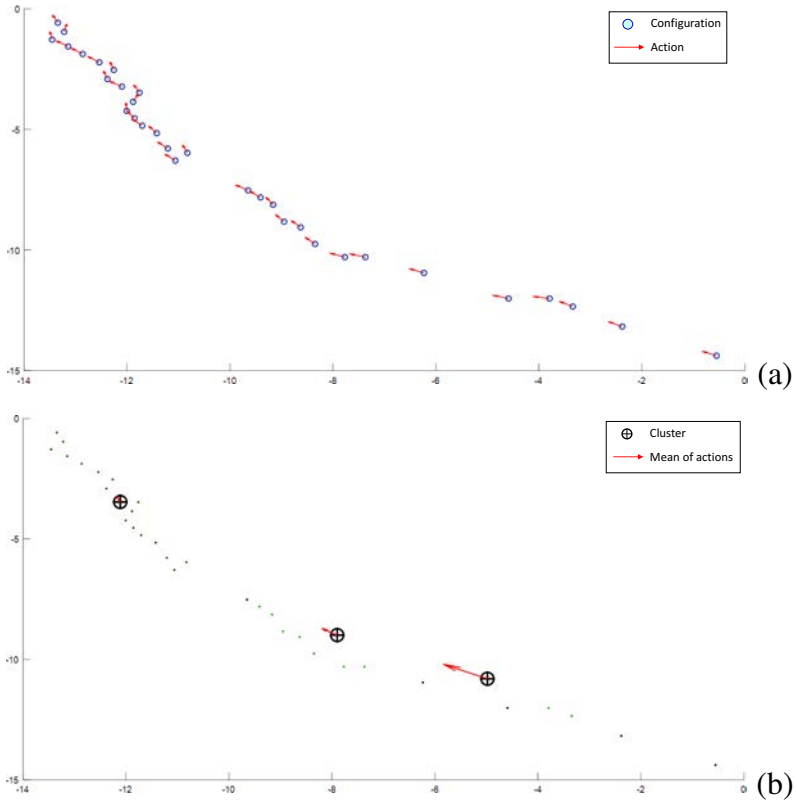


Fig. 4.17 Clustering the explored configurations in the lane-changing scenario. (a) shows all-new exploration by the learner through one step and (b) shows the clustered configurations and the corresponding mean action value to them.

clusters of the reference FP-M (circles in gray) and the newly learned ones that are appended to the reference model (circles in yellow) in two different examples when L aims to overtake the dynamic object V.

The corresponding Π is updated by adding new rows and columns that represent the newly learned configurations as shown in Fig. 4.20-(b) and Fig. 4.21-(b) and the transition matrix Π of the reference FP-M are shown in Fig. 4.20-(a) and Fig. 4.21-(a). Comparing sub-figures (a) and (b) in each Fig. 4.20 and Fig. 4.21 shows how the transition matrix of the FP-M are expanded after L has explored and learned new situations allowing to predict the environmental dynamics in the future better and consequently select effective actions. Such an incremental learning process under AIn endows L with the capability of understanding the best set of actions it should perform to avoid surprising states.

L adopting the proposed AIL method has higher successful movements than IRL, SL, Q-learning, and DQN, as depicted in Fig. 4.22. Two factors directly affect the success of the learner travel in each episode: the probability of going out of the boundary and the collision probability. As we mentioned earlier, each episode includes ten steps (ten full

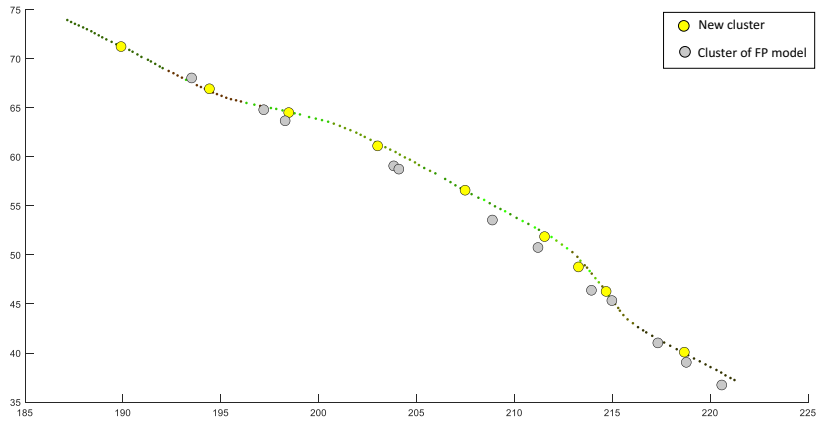


Fig. 4.18 This figure shows the incremental learning of the model through the online learning phase. The gray circles show the clusters that belong to the FP-M, and the yellow circles present the newly added clusters to the AFP-M, which is learned through the exploration.

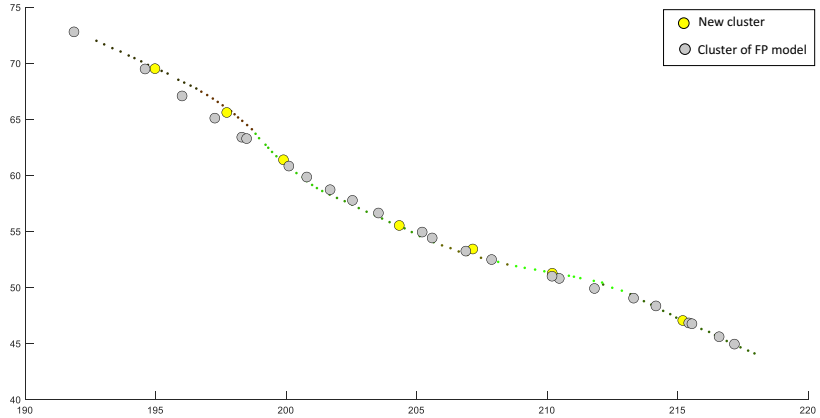


Fig. 4.19 This figure shows another example of the incremental learning of the model through the online phase. The gray circles show the clusters that belong to the FP-M, and the yellow circles present the newly added clusters to the AFP-M, which is learned through the exploration.

paths). Obviously, with two factors decreasing at each step, the growth of the success steps led to an increase in the successful travel in each episode. By way of explanation, during the exploration, the model minimizes the FE measurement at the DL (F_{DL}) at time k, which causes the resemblance between predictions and evidence at the CL.

In total, by optimizing the global FE defined in (4.25) in the unseen situations, the learner can manage to avoid a collision with another agent or going out of the boundary. Fig. 4.23 shows the collision probabilities in each episode. We observe that the collision probability decreases as the number of episodes increases.

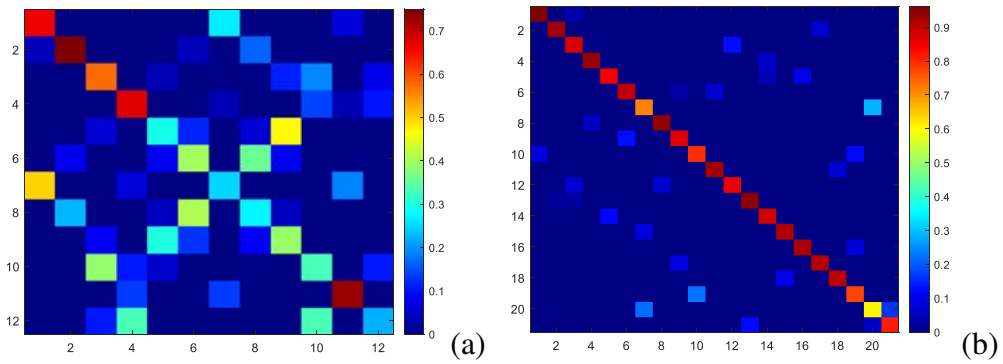


Fig. 4.20 This figure is based on Fig. 4.18. (a) describes the transition matrix related to the FP-M in case of lane-changing from the right side that includes 12 cells, and (b) shows the transition matrix with 21 cells after learning a new set of clusters (yellow circles in Fig. 4.18) that explains how the number of clusters increases in each online learning step.

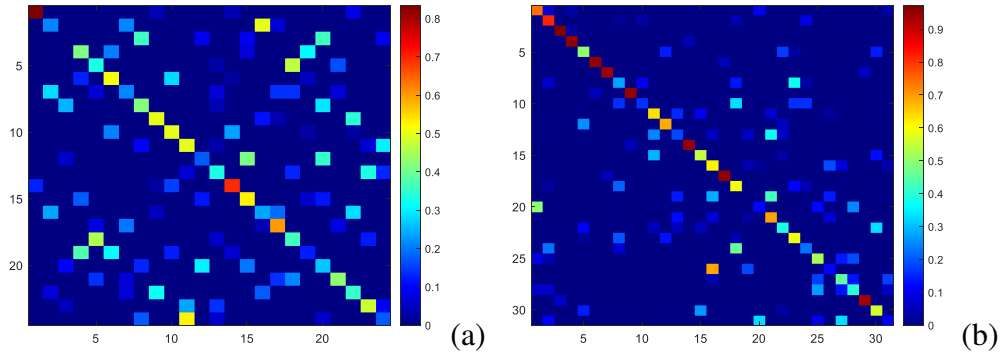


Fig. 4.21 This figure is based on Fig. 4.19. (a) describes the transition matrix related to the FP-M in case of lane-changing from the left side that includes 24 cells, and (b) shows the transition matrix with 31 cells after learning a new set of clusters (yellow circles in Fig. 4.19) that explains how the number of clusters increases in each online learning step.

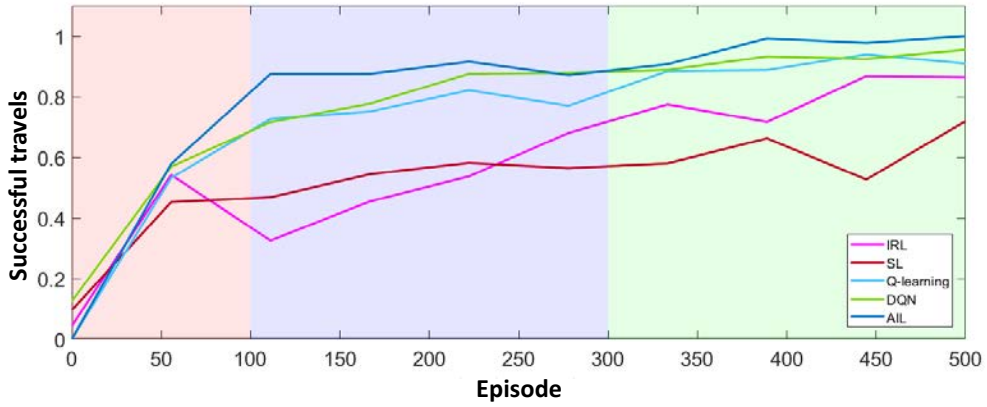


Fig. 4.22 Success rate to accomplish the task.

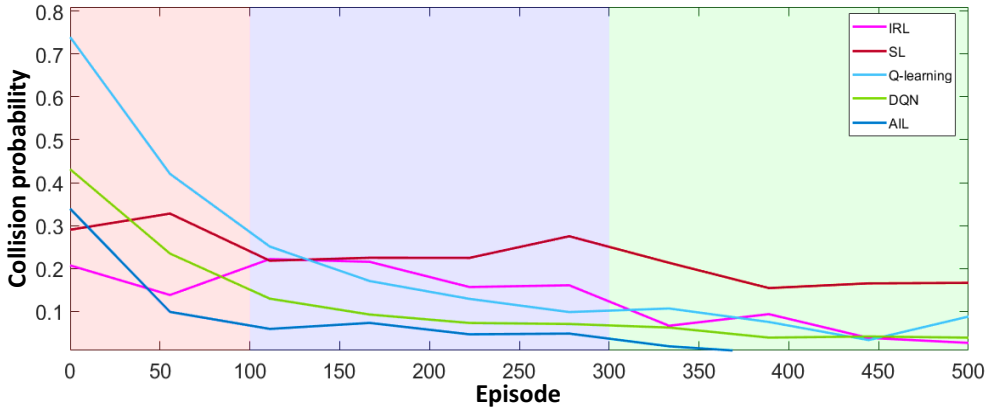


Fig. 4.23 Analysis of the learning process: collision probability.

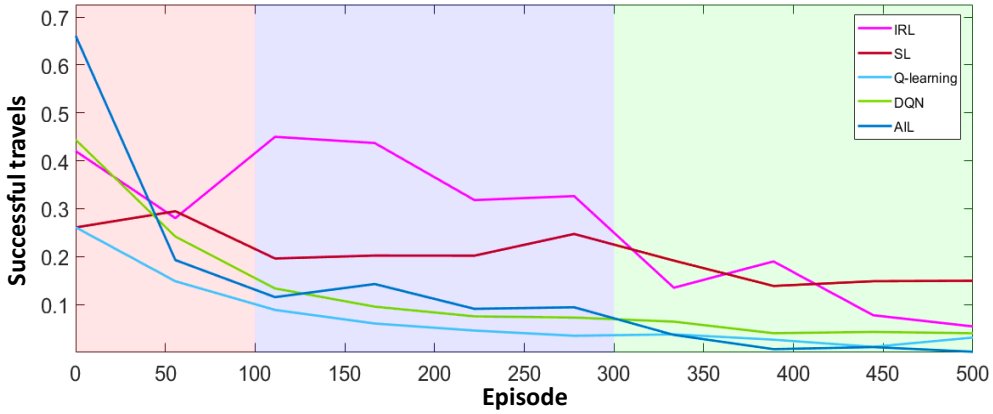
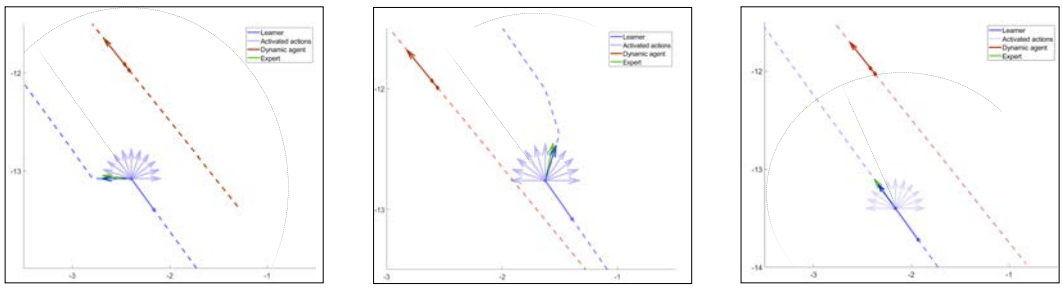


Fig. 4.24 Analysis of the learning process: going out of boundary probability.

Moreover, Fig. 4.24 presents the probabilities of going out of the boundary that starts with 62% and, during the learning, dramatically declines to 0%. Fig. 4.23 and Fig. 4.24 justifies the L behavior in Fig. 4.22.

The experimental results demonstrate that the proposed method enabled L to learn better driving skills than other RL methods. Integrating IL with AIn gives L a prior driving experience, accelerating the learning rate and improving the driving policy. The presented quantitative results prove that the proposed method improves the IL using expert demonstrations by taking advantage of sub-optimal reference data (exploitation) and dynamically involving FE measurements at both DL and CL to minimize the distinction between the SM and AFP-M.

Furthermore, qualitative results show the ability to manage critical situations. Fig. 4.25 shows some representative cases of different scenarios. L's activated motion, the dynamic candidate motions, and the expert driving action (the ground truth) are displayed with blue, grey, and green arrows, respectively. The associated probabilities to the candidate motion



(a) (b) (c)

Fig. 4.25 In three cases, the learner agent's preference (blue arrow) is similar to the expert behavior (green arrow) in the same situation. Also, the agent has learned to keep a safe distance (gray dashed line) with another dynamic agent.

Table 4.1 This table shows the probability of actions selection by learner agent in Fig. 4.25 where it changes the lane to the left and right side, also where it keeps the lane. For each case, $a1$, $a7$, and $a4$ is the selected action, respectively, with the highest probability.

probable actions		a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11
selection probability(%)	lane changing - to left	64.843	21.021	8.008	2.374	2.152	1.042	0.184	0.133	0.122	0.087	0.034
	lane changing - to right	0.021	0.040	0.089	2.201	2.325	18.191	48.717	22.052	3.264	1.905	1.195
	lane keeping	0.682	1.736	4.158	89.30	3.616	0.339	0.135	0.014	0.011	0.003	0.002

are depicted in Table. 4.1. In each case of decision (lane-keeping, change to the left, and change to the right), the most likely motion to the expert is selected, which has the highest probability than other candidates. Table. 4.1 shows the probability percentage of the activated actions in all three cases.

4.5.5 Imitation Loss

Our goal is to find the best set of actions that minimize imitation loss in terms of FE measurements. Fig. 4.26 shows that the normalized global FE ($\mathscr{G}$) drops capably in less than 50 training episodes, and after 200 episodes, its value continues to decrease below 0.1. Moreover, Fig. 4.28 shows the $\mathscr{G}$ performance considering different L's preference, i.e., to keep following the other dynamic agent V, overtake from the left side or overtake from the right side.

Two main factors affect the global FE: the motion distinction at time k and the divergence at time $k+1$ after performing a specific action by the L agent. Fig. 4.29 illustrates the imitation loss during the online active learning phase. We prove that our method can minimize the motion distinction (F_{DL}), which is under the control of action selection at each

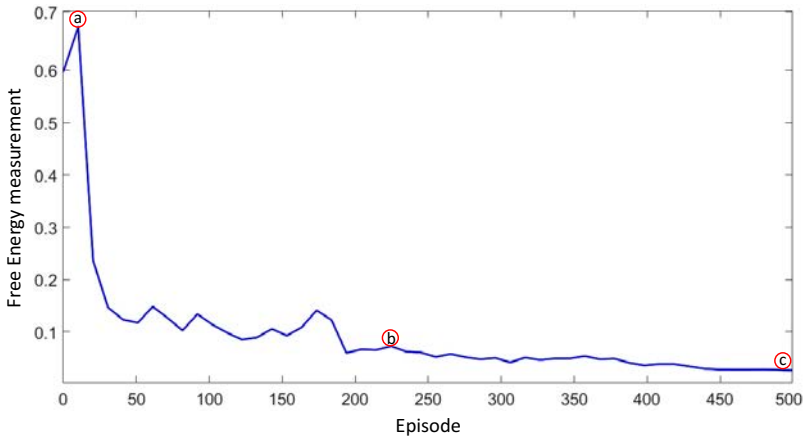
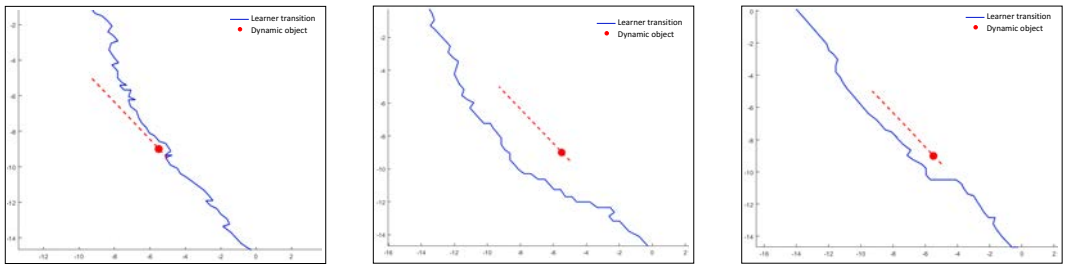


Fig. 4.26 Global Free Energy measurement $\mathscr{G}$. The red circles show the FE measurement through three slots of learning: (a) shows the beginning of training when the learner tries to experience the new action, in (b) the FE is declined cause improving the action selection, and at (c) learner could decrease the distinction with the expert configurations. Fig. 4.27 shows the three trajectories based on the mentioned measurements.



(a) (b) (c)

Fig. 4.27 This figure shows three trajectories based on the selected FE measurements in Fig. 4.26. In (a), the learner can not balance exploration and exploitation yet. By decreasing imitation loss and improving the explored actions, the learner finishes the travel by taking fewer actions as demonstrated in (b), and (c) shows a successful travel with suitable actions concerning the dynamic object's situation.

instant. Further, improving the action selection process leads to minimizing the divergence (F_{CL}) between prediction and evidence. Therefore, by minimizing the imitation loss in both cases, L learns to maximize the likelihood with the E behavior and overtakes the unobserved situation. In addition, Fig. 4.30 shows that the proposed AIL is capable of achieving higher imitation rates than other learning methods.

Fig. 4.31-(a)-(b) presents the performance of the proposed method (AIL) in terms of success rate, collision rate, and out of boundary rate during training and testing, respectively. Also, Fig. 4.31-(a)-(b) provides a comparison with other methods. It is shown that the

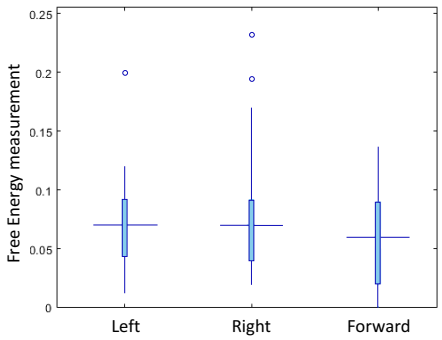


Fig. 4.28 Free Energy measurement in three cases: lane-changing from left, lane-changing from right, and lane-keeping.

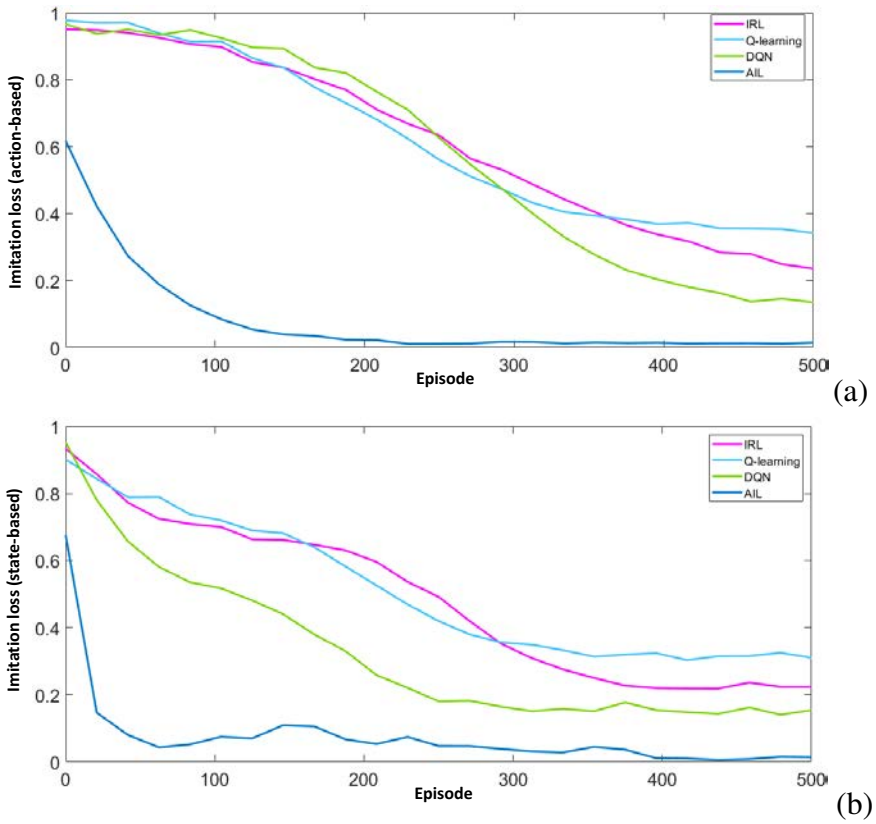


Fig. 4.29 Analysis of the imitation process after 500 training episodes ($5k$ path): (a). Motion distinction. This figure shows the motion difference between the learner and the expert agent through the online learning active learning phase at time k. (b) Divergence measurement. This figure shows the divergence between the learner and expert agent state after taking action at time $k+1$.

proposed method (AIL) performs best among all methods (during training and testing), which is attributed to the effectiveness of the decision-making while dealing with dynamic

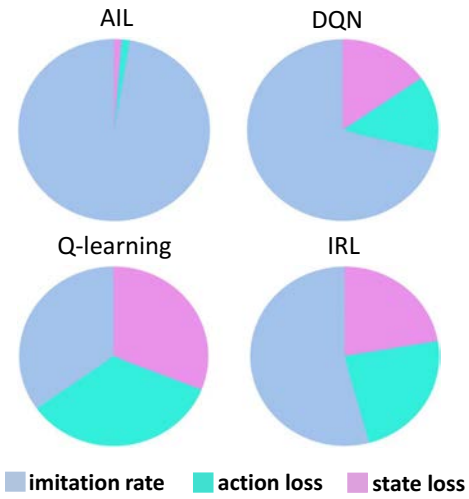


Fig. 4.30 Imitation loss: comparing different learning techniques.

Table 4.2 Results after 500 training episodes. * In the SL method, there are no optimal expert demonstrations.

	AIL	DQN	Q-learning	SL	IRL
train					
success %	**97.21**	92.12	86.51	60.00	92.10
collision%	**0.93**	3.89	9.98	21.08	2.57
out of boundary%	**1.86**	4.00	3.51	18.92	5.33
imitation rate	**0.974**	0.712	0.349	*	0.542
imitation loss	**0.026**	0.288	0.651	*	0.458
action loss	**0.013**	0.135	0.341	*	0.235
state loss	**0.012**	0.153	0.310	*	0.223
number of taken action(mean)	**25**	44	42	64	60
test					
success%	**97.96**	82.01	76.13	70.50	78.92
collision%	**0.70**	6.33	15.30	19.83	9.54
out of boundary%	**1.34**	11.66	8.57	9.67	11.54

changes in the environment that improve the success rate by preventing going out of boundary and avoiding collisions.

Besides, during testing, results showed that by $5k$ training episodes, the agent can change-line to overtake the other dynamic agent in the environment effectively while other methods still have high collision probabilities, as shown in Fig. 4.32. Correspondingly, Table. 4.2 summarises the performance metrics and presents the comparison with other methods.

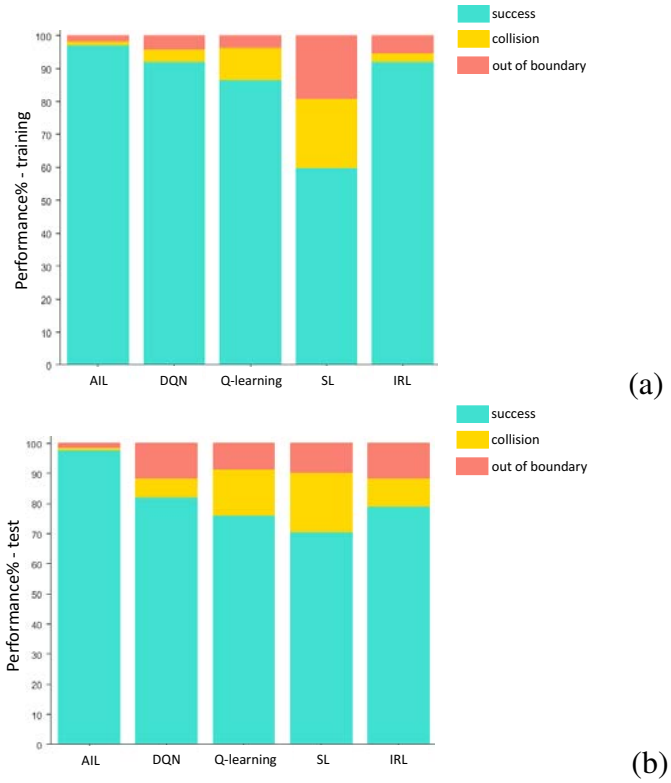


Fig. 4.31 (a) shows the training results after $5k$ iterations. AIL has less training loss (collision and going out of boundary) than other methods, and it causes more training success percentage. (b) demonstrates the testing results through the 500 paths. The testing paths have different start positions than the training, and the dynamic object moves with different velocities during the training phase. It shows that the trained agent by AIL can achieve a high success percentage in the new environment.

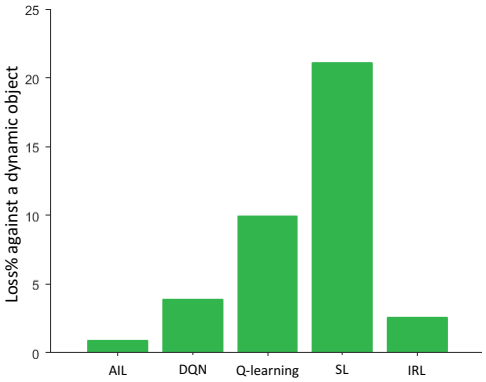


Fig. 4.32 The result of overtaking loss from a dynamic object during changing-line.

4.6 Model II - Employ modified MJPF to Active Inference

During the online learning phase, L utilizes a hybrid mechanism combining IL with AIn to acquire efficient exploratory actions and adapt quickly to new tasks. Particularly, the Markov Jump Particle Filter (MJPF) is implemented to perform joint predictions of configurations and GSs during the online learning phase (see Fig. 4.33). The incremental learning procedure involves five main steps: 1)prediction and perception, 2) action selection, 3)transition model update, 4) FE measurement, and 5) action updates. The logic of the AIn approach is reported in **Algorithm 2**.

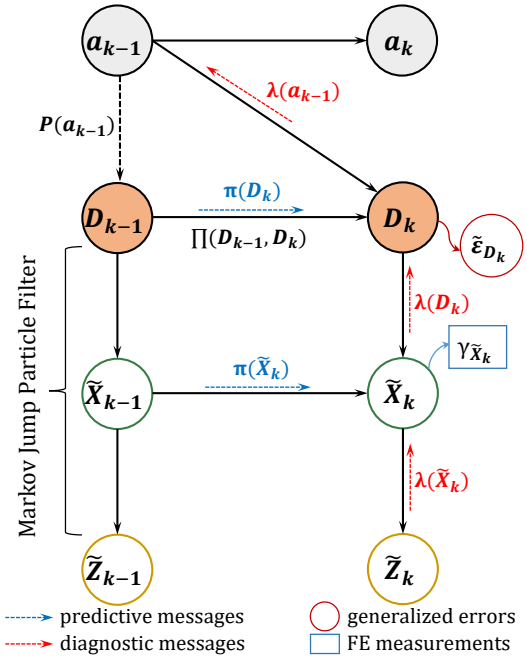


Fig. 4.33 Active First-Person model

4.6.1 Prediction and Perception

A Modified Markov Jump Particle Filter (M-MJPF) is employed that uses a combination of Particle Filter (PF) and Kalman Filter (KF) to provide various probabilistic inference modes, namely, predictive inference and diagnostic inference. In the former mode, predictive messages holding beliefs in hidden states at multiple levels are propagated in a top-down manner. In the latter mode, diagnostic messages are fed back in the opposite direction from bottom-to-up of the hierarchy to update beliefs in hidden variables given a sequence of observations and calculate FE and GEs. First, PF propagates a set of N particles equally weighted using a specific row ($\pi(\mathrm{D}_k)$) in Π_D as a proposal distribution, such that,

$\{\mathrm{D}_{k,n}{\sim}\pi(\mathrm{D}_k), \mathrm{W}_{k,n}{=}\frac{1}{N}\}$. Then, for each particle n, a KF is employed to predict the GS as pointed out in 4.9. The predicted GS depends on the prediction done at the higher level, which can be written as a condition probability $\mathrm{P}(\tilde{\mathrm{X}}_\mathrm{k}|\tilde{\mathrm{X}}_{\mathrm{k}-1}, \mathrm{D}_{\mathrm{k,n}})$. The posterior probability associated with the predicted configuration is given as follows:

$$\pi(\tilde{\mathrm{X}}_k) = \mathrm{P}(\tilde{\mathrm{X}}_k, \mathrm{D}_{k,n}|\tilde{Z}_{k-1}) = \int \mathrm{P}(\tilde{\mathrm{X}}_k|\tilde{\mathrm{X}}_{k-1}, \mathrm{D}_{k,n})\lambda(\tilde{\mathrm{X}}_{k-1})d\tilde{\mathrm{X}}_{k-1}, \quad (4.28)$$

where $\lambda(\tilde{\mathrm{X}}_{k-1}) = \mathrm{P}(\tilde{\mathrm{Z}}_{k-1}|\tilde{\mathrm{X}}_{k-1})$. Accordingly, a diagnostic message backward propagated from bottom-to-up of the hierarchy once a new sensory signal $\tilde{\mathrm{Z}}_k$ is measured can be exploited to update the posterior $\mathrm{P}(\tilde{\mathrm{X}}_k, \mathrm{D}_{k,n}|\tilde{Z}_k)$ according to:

$$\mathrm{P}(\tilde{\mathrm{X}}_k, \mathrm{D}_{k,n}|\tilde{Z}_k) = \pi(\tilde{\mathrm{X}}_k)\lambda(\tilde{\mathrm{X}}_k), \quad (4.29)$$

where $\lambda(\tilde{\mathrm{X}}_k){=}\mathrm{P}(\tilde{\mathrm{Z}}_k|\tilde{\mathrm{X}}_k)$. Likewise, the likelihood message $\lambda(\mathrm{D}_k)$ propagated towards the top level can be used to change the belief in the hidden discrete states by updating the particles' weight according to:

$$\mathrm{W}_{k,n}{=}\mathrm{W}_{k,n}{\times}\lambda(\mathrm{D}_k), \quad (4.30)$$

where $\lambda(\mathrm{D}_k){=}\lambda(\tilde{\mathrm{X}}_k)\mathrm{P}(\tilde{\mathrm{X}}_k|\mathrm{D}_k)$ is a probability discrete distribution and $\mathrm{P}(\tilde{\mathrm{X}}_k|\mathrm{D}_k){\sim}\mathcal{N}(\tilde{\mu}_{\mathrm{D}_k}, \Sigma_{\mathrm{D}_k})$. After updating the weights, PF uses the sequential importance resampling (SIR) to assign new weights to be used in a successive instant.

4.6.2 Action Selection

The action selection process is based on the updated particles' weights to decide whether to exploit actions by imitating the learned configurations or to explore new actions that minimize FE (maximize rewards) in the future. L decides between exploration and exploitation by using two parameters, namely, the exploration rate (ε) and a varying threshold (t) which is defined based on a trial-and-error process. The exploration rate is defined as:

$$\varepsilon_k = 1 - \alpha_k, \quad (4.31)$$

where α_k is the largest weight among all the N particles measuring the likelihood between the current L configuration and the memorized configurations, such that:

$$\alpha_k = \max_n W_{k,n}, \tag{4.32}$$

where $0 \leq \alpha_k \leq 1$. So, if α_k is near 1, ε_k becomes very low, which means that current observation matches L's expectation and so it can exploit the same actions performed by E.

Thus, the action selection process depends on the decision made by L whether to explore or exploit according to:

$$a_k \sim \begin{cases} \mu_V^{\mathrm{D}_k^\beta} = \arg\max_{a_k} \mathrm{Q}(\mathscr{A}, \mathrm{D}_k^\beta), \text{ if } \varepsilon < t \text{ (exploitation)}, \\ \textit{random} \text{ from } \mathscr{A}^+, \text{ if } \varepsilon \geq t \text{ (exploration)}, \end{cases} \tag{4.33}$$

where a_k are the active states (i.e., actions) realizing the top level of the AFP-M, $\mathscr{A} = \{\mathscr{A}^{\mathrm{E}}, \mathscr{A}^+, \mathscr{A}^\dagger\}$, where $\mathscr{A}^{\mathrm{E}} = \{a_1^{\mathrm{E}}, a_2^{\mathrm{E}}, \ldots, a_{\mathrm{Y}}^{\mathrm{E}}\}$ is a set of actions performed by E and encoded in SM that L aims to imitate during the exploitation (a greedy policy). Moreover, $\mathscr{A}^+ = \{a_1, a_2, \ldots, a_8\}$ is a set of predefined actions realizing eight different directions[2] used during the exploration and $\mathscr{A}^\dagger$ is the set of new explored actions after performing clustering (as we discuss later in this section). In addition, D_k^β is the most similar learned configuration to the observed one, and β is the particle's index with the maximum weight associated with (4.32) defined as:

$$\beta = \arg\max_n (W_{k,n}) \tag{4.34}$$

4.6.3 Transition Model update

The dynamic transitions among the learned configurations at the top level of the hierarchy are encoded in a transition matrix (Π) that can be learned by estimating the transition probabilities $\mathrm{P}(\mathrm{D}_{k+1}|\mathrm{D}_k)$ as:

$$\Pi = \begin{bmatrix} \mathrm{P}(\mathrm{D}_1|\mathrm{D}_1), & \ldots, & \mathrm{P}(\mathrm{D}_1|\mathrm{D}_\mathrm{M}) \\ \vdots & \ddots & \vdots \\ \mathrm{P}(\mathrm{D}_\mathrm{M}|\mathrm{D}_1), & \ldots, & \mathrm{P}(\mathrm{D}_\mathrm{M}|\mathrm{D}_\mathrm{M}) \end{bmatrix}, \tag{4.35}$$

where $\sum_m^{\mathrm{M}} \mathrm{P}(\mathrm{D}_p|\mathrm{D}_m) = 1$ such that $p, m \in \mathrm{M}$. During exploration, L saves the newly explored configurations D_k^+ along with the performed actions $a_k^+ \in \mathscr{A}^+$ in a set $\mathscr{C}$. After completing a certain number of experiences that requires τ_e episodes, L clusters all the pairs $[\mathrm{D}_k^+, a_k^+]$ saved in $\mathscr{C}$ by employing the GNG that outputs a set $\mathrm{D}^\dagger = \{\mathrm{D}_1^\dagger, \mathrm{D}_2^\dagger, \ldots, \mathrm{D}_{\mathrm{M}^\dagger}^\dagger\}$ of $\mathrm{M}^\dagger$ clusters

[2]The eight directions are North, South, East, West, North-West, North-East, South-East, South-West.

representing the new configurations ($\mathrm{D}_m^\dagger \in \mathrm{D}^\dagger$) and a set $\mathscr{A}^\dagger$ representing the corresponding actions. The new configurations and actions in sets $\mathrm{D}^\dagger$ and $\mathscr{A}^\dagger$ can be appended incrementally to a probabilistic table (Q) that can be exploited in the future.

In addition, L updates the transition model defined in (4.35) during *exploration* by adding new rows and columns which are related to the novel configurations learned incrementally as follows:

$$\Pi^* = \begin{bmatrix} \Pi & \Pi^\dagger \\ \Pi^\ddagger & \Pi^{\dagger\dagger} \end{bmatrix}, \tag{4.36}$$

where $\Pi^\dagger \in \mathbb{R}^{\mathrm{M},\mathrm{M}^\dagger}$, $\Pi^\ddagger \in \mathbb{R}^{\mathrm{M}+\mathrm{M}^\dagger,\mathrm{M}}$, $\Pi^{\dagger\dagger} \in \mathbb{R}^{\mathrm{M}+\mathrm{M}^\dagger,\mathrm{M}+\mathrm{M}^\dagger}$ encode the new probability transitions corresponded to the new configurations (i.e., the output of the GNG that takes in input $\mathscr{C}$) appended to the updated transition matrix (Π^*). Moreover, during *exploitation*, L modifies and update the transition matrix (Π) by decreasing the probability of transiting from D_{k-1} to D_k after selecting action a_{k-1} as:

$$\Pi(\mathrm{D}_{k-1},\mathrm{D}_k) = \Pi(\mathrm{D}_{k-1},\mathrm{D}_k) + \tilde{\varepsilon}_{\mathrm{D}_k}, \tag{4.37}$$

where $\tilde{\varepsilon}_{\mathrm{D}_k}$ is the GE at the discrete level that plays the role of a dynamic force to evaluate the L's transitions, which is defined as:

$$\tilde{\varepsilon}_{\mathrm{D}_k} = \lambda(\mathrm{D}_k) - \pi(\mathrm{D}_k). \tag{4.38}$$

4.6.4 Free Energy Measurement

L evaluates its accomplishments by comparing predictions with sensory input to the AFP-M in terms of FE computation. We aim to find a policy where L's behavior matches the reference demonstrations. For this purpose, our objective is to minimize the divergence between what L is expected to observe after taking a certain action and what is the actual observation under both exploration and exploitation. Hence, the expected FE at the CL is calculated based on the Bhattacharyya distance [76] to evaluate how much the observation supports predictions:

$$\Upsilon_{\tilde{\mathrm{X}}_k} = \mathscr{D}_B\Big(\pi(\tilde{\mathrm{X}}_k,\lambda(\tilde{\mathrm{X}}_k))\Big) = -\ln\Big(\mathscr{BC}\big(\pi(\tilde{\mathrm{X}}_k),\lambda(\tilde{\mathrm{X}}_k)\big)\Big), \tag{4.39}$$

where $\mathscr{BC}(.) = \int \sqrt{\pi(\tilde{\mathrm{X}}_k)\lambda(\tilde{\mathrm{X}}_k)}d\tilde{\mathrm{X}}_k$ is the Bhattacharyya Coefficient.

Algorithm 3 Active Inference

Input: TM, **D**, Q ← Transition Matrix, Configurations, QTable
for $k = 1$ to K ← Time evolution **do**
  for $n = 1$ to N ← Particles **do**
    if $k == 1$ ← Initial iteration **then**
      Initialization:
      $P(\tilde{X}_0) \sim \mathcal{N}(\mu_{\tilde{X}_0}, \Sigma_{\tilde{X}_0})$ ← prior distribution
      Sample $\tilde{X}_k \sim P(\tilde{X}_0)$
      $P(D_0) = \mathcal{U}\{1, |D_m|\}$ ← uniform distribution with pmf= $\frac{1}{|D_m|}$
      Sample $D_k \sim P(D_0)$
      $W_{k,n} = \frac{1}{N}$ ← particle weight
    else
      $D_{k,n} \sim TM(D_{k-1,n})$ ← proposal from transition matrix
      Prediction at the continuous level:
      $\tilde{X}_k = A\tilde{X}_{k-1} + B\dot{\mu}_{D_{k,n}} + w_k$ ← predicted distance btw L and O
      $\Sigma_{\tilde{X}} = A\Sigma_{\tilde{X}_{k-1}}A^\intercal + \Sigma_{D_{k,n}}$ ← the predicted covariance
      $\pi(\tilde{X}_k) \sim \mathcal{N}(\mu_{\tilde{X}_k}, \Sigma_{\tilde{X}_k})$ ← Predictive msg
    end if
    Receiving observation $\tilde{Z}_k$ **(relative distance btw L and O)**
    $\lambda(\tilde{X}_k) = P(\tilde{Z}_k|\tilde{X}_k)$
    $\mathscr{Y}_k = \tilde{Z}_k - H\tilde{X}_k$ ← Kalman Innovation
    $\mathscr{E}_k = H^{-1}\mathscr{Y}_k$ ← Generalized Errors
    $\lambda(D_k) = \lambda(\tilde{X}_k)P(\tilde{X}_{k,n}|D_{k,n})$
    FE measurement:
    $\Upsilon_{\tilde{X}_k} = -\ln\Big(\mathscr{BC}\big(\pi(\tilde{X}_k), \lambda(\tilde{X}_k)\big)\Big)$
    Update:
    $W_{k,n} = W_{k,n} \times \lambda(D_k)$ ← updated weight
    SIR resampling
    $W_{k+1,n} = \frac{1}{N}$
  end for
  $[a_k, \mathscr{C}]$ = `ACTION SELECTION`$(W_{k,n})$ ▷ the effect of a_k will be evaluated at $k+1$
  Update Q using (4.40) ← action update
end for
$[D^\dagger, \mathscr{A}^\dagger]$ = `GNG`$(\mathscr{C})$
Q.$append(D^\dagger)$
Q.$append(\mathscr{A}^\dagger)$
$TM^\dagger$ = TM.$append(D^\dagger)$ according to (4.37)

Algorithm 4 Action selection

function ACTION SELECTION($W_{k,n}$)
  $\beta = \arg\max_n W_{k,n}$ ← index of max
  $\alpha = \max_n W_{k,n}(D_{k,n})$ ← value of max
  The corresponded configuration to β presents the activated reference configuration
  $D_k^{active} = D_{k,\beta}$.
  ρ ← Threshold for adding new configuration
  $\varepsilon = 1 - \alpha$ ← exploration rate
  if $\varepsilon < \rho$ **then**
    $a_k \sim \mu_V^{D^{k,\beta}} = \arg\max_{a_k} Q(\mathscr{A}, D_{k,\beta})$ ← exploitation
  else
    $a_k^+ \sim$ random from $\mathscr{A}^+$ ← exploration
    save $[D^+, a_k^+]$ in $\mathscr{C}$
  end if
  return $a_k, \mathscr{C}$
end function

L believes that a certain action allows it to imitate the E's behavior during exploitation correctly or allows it to approach towards the E's reference vocabulary as soon as possible during the exploration

4.6.5 Action Update

L aims to improve the action selection that minimizes the cumulative FE (C-FE). L adapts the action selection process by updating Q based on the FE expressed in (4.39), also since Q is a probabilistic table, it can be rewritten in probabilistic form as follows:

$$\mathrm{Q}^{*} = (1-\eta)\mathrm{P}(a_{k-1}|\mathrm{D}_{k-1}) + \eta\left[(1-\bar{Y}_{\tilde{X}_k}) + \gamma \max_{a_k} \mathrm{P}(a_k|D_k)\right]. \tag{4.40}$$

where η is the learning rate that controls how quickly the learning agent adopts to the explorations imposed by the environment, $\bar{Y}_{\tilde{X}_k}$ is the normalized FE measurement with a range from 0 to 1, and γ is a discount factor as in the general case of RL algorithms.

Modifying and improving the beliefs under AIn allows L to motivate the exploratory actions to seek to extend its knowledge and resolve uncertainty in a Bayesian hierarchical structure. Furthermore, the evaluation relies on the dynamic forces computed using the GEs that can be treated as self-information to reach equilibrium.

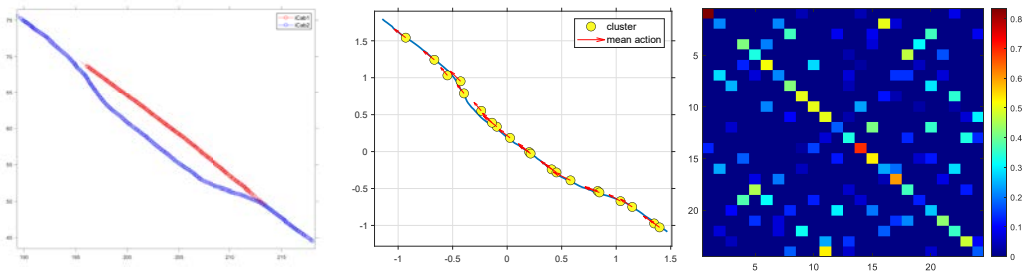
4.7 Simulation and Performance Evaluation - Model II

4.7.1 Experimental Data Set

The proposed framework is validated using a real dataset consisting of multisensorial information collected from two AVs, 'iCab 1' and 'iCab 2' [91]. The vehicle's positional information and the corresponding velocities are obtained from the odometry module to consider the lane-changing scenario when *iCab*2 needs to change its home lane to overtake *iCab*1 without collision.

4.7.2 Offline Learning Phase

This section shows the process of learning the SM from sensory data while the NFF is used as an initial filter employed on the collected data. NFF outputs the GEs, which can be clustered using GNG that outputs a set of discrete clusters representing the discrete regions of the trajectories generated by E and O. The collected sensory data are processed as explained in



(a) (b) (c)

Fig. 4.34 Learning the Situation model. a) iCab2 overtakes iCab 1 from the left side, b) Clustering of GEs, c) corresponding Transition Matrix (Π).

4.3.1. The SM represents 24 joint clusters $(D = D_1, D_2, ..., D_{24})$ that encode the dynamic interaction between two participants in the environment (see Fig. 4.34).

Additionally, in the offline learning phase, Q is initialized from the FP-M (explained in 4.3.2). Thus it contains 24 learned configuration based on the observed interactions in SM and their associated actions such as:

$$\mathrm{Q} = \begin{array}{c} \\ a_1 \\ a_2 \\ \vdots \\ a_{24} \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{D}_1 & \mathrm{D}_2 & \dots & \mathrm{D}_{24} \end{array} \\ \begin{bmatrix} \frac{1}{24} & \frac{1}{24} & \cdots & \frac{1}{24} \\ \frac{1}{24} & \frac{1}{24} & \cdots & \frac{1}{24} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{24} & \frac{1}{24} & \cdots & \frac{1}{24} \end{bmatrix} \end{array}. \tag{4.41}$$

4.7.3 Online Learning Phase

During the online learning phase, L develops Q during $5k$ training episodes in a simulated environment, where the direction of the dynamic object (i.e., an obstacle (V) is changing. We evaluate the performance of the proposed method and compare it with three learning algorithms, conventional Q-learning, inverse reinforcement learning (IRL) when optimal expert data is available, and self-learning (SL) in RL context without access to the expert data.

The action selection procedure plays an essential role in decreasing the divergence between prediction and observation by balancing the exploratory and exploitative movements, which is one of the most challenging tasks in RL. Modifying the action selection through adapting novel experiences minimizes the exploration rate that causes lower FE in the future. Fig. 4.35 demonstrates how increasing the exploitation rate (in order to decrease the exploration) reduces FE. Furthermore, Fig.4.36 illustrates how using GNG to cluster the newly

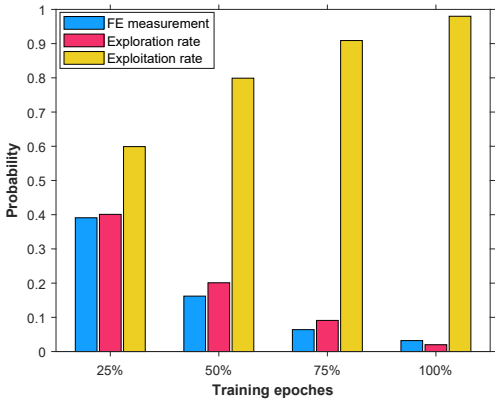


Fig. 4.35 The graph illustrates the exploitation and exploration rates after each training quarter and their effects on the FE.

experienced configurations during the online learning phase decreases the FE measurement satisfying and accelerates the convergence between the expectation and evidence rather than applying the approach without employing GNG.

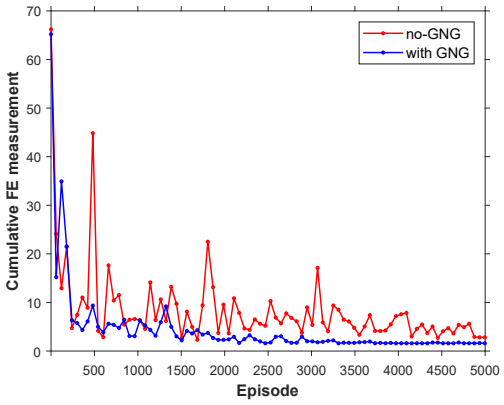
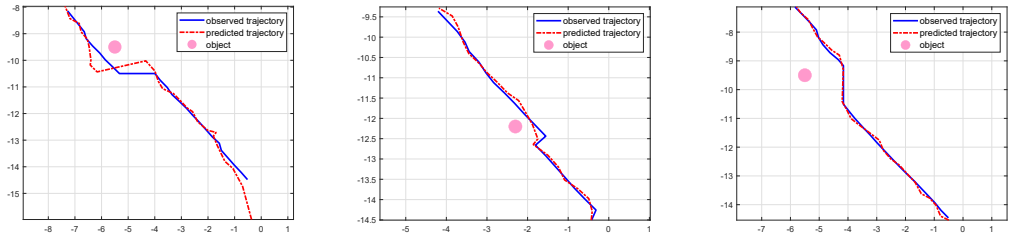


Fig. 4.36 The figure shows how cumulative FE measurement converges properly by taking advantage of GNG in the online phase.

Our goal is minimizing the imitation cost in terms of FE, Fig. 4.37 shows three trajectories based on three different times while learning the model (i.e., after a quarter, after a half, and after full training). In Fig. 4.37-(a), L can not balance the exploration and exploitation yet. By improving the action selection policy and decreasing FE, L can finish the travel by performing more optimized actions than previous episodes (Fig. 4.37-(b)), and Fig. 4.37-(c) shows a successful mission with the suitable actions concerning the dynamic object's situation. Furthermore, Fig. 4.37 shows that L learns lane-changing to overtake the object

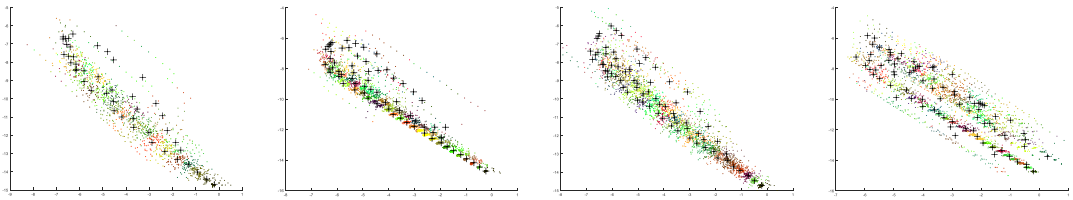


(a) (b) (c)

Fig. 4.37 In (a), the learner experiences new actions by exploration. In (b), by balancing the exploration and exploitation, the learner reduces the distinction between the observation (i.e., the trajectory that it is following) and prediction (i.e., the trajectory that it is supposed to follow), and (c) shows how the learner minimizes the divergence and performs suitable actions concerning the dynamic object during training.

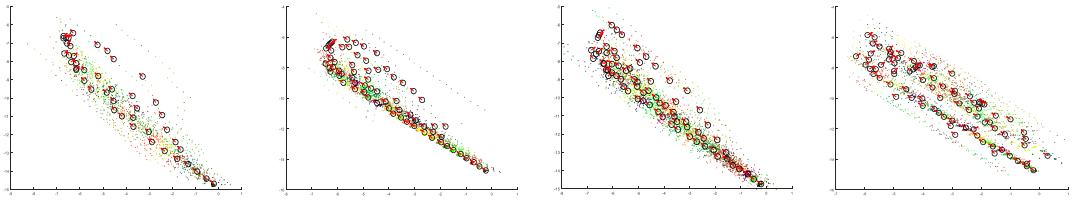
from both the right and left side through exploratory behavior, also Q and the corresponding transition matrix are expanded incrementally (as discussed in 4.6.2).

While L is trained to have an opportunity to gain new experiences from the environment's dynamic, it follows the expected predictions to accomplish its task. L records newly explored configurations along with the performed actions in the exploration stage. Then it clusters the newly recorded pairs (as discussed in 4.6.2) to calculate the mean and the derivative of the corresponding clusters in order to have probabilistic data in line with the FP-M. Fig. 4.38 describes the clustering stage during the online learning phase and shows the incremental learning process. Fig. 4.38-(a)-(b)-(c)-(d) shows how the model evolves incrementally during training by adding the novel learned configurations (or clusters) after applying GNG during the online learning phase. Fig. 4.39-(a)-(b)-(c)-(d) demonstrates the mean action value of each cluster, and Fig. 4.40-(a)-(b)-(c)-(d) illustrates the corresponding transition matrices which are expanding during the training stage.



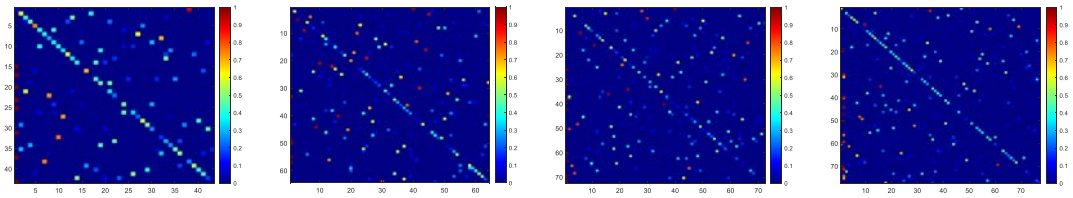
(a) (b) (c) (d)

Fig. 4.38 Clustering process of the explored configurations and actions during the online phase. This figure shows the output of GNG (i.e., clusters) after each training quarter.



(a) (b) (c) (d)

Fig. 4.39 The associating mean action to each cluster.



(a) (b) (c) (d)

Fig. 4.40 The sub-figures illustrate how in each training quarter the transition matrix evolves by adding incrementally the new clustered configurations.

We prove the proposed approach can minimize the FE through the online learning phase under the effect of action selection modification at each instant. Further, correcting the actions and updating Π cause convergence between prediction and evidence that leads L to maximize the reward amount $(1-\bar{Y}_{\tilde{X}_k})$, as pointed out in (**??**). L learns to maximize the likelihood with E behavior and overtake the unobserved situation (i.e., lane-changing from the right side of a dynamic object). Fig. 4.41 shows that the proposed method achieves a higher cumulative reward in a shorter time than other learning methods. Results show that

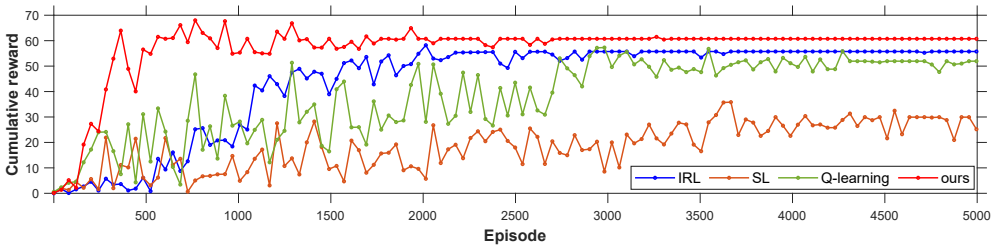


Fig. 4.41 Performance comparison in terms of cumulative reward.

with $5k$ training episodes, the agent can effectively overtake the other dynamic agent in the environment, while this is still challenging for other methods.

4.8 Conclusion

This chapter introduced two system models for autonomous driving in a dynamic environment. In Model I - Active Inference integrated with Imitation Learning, a framework is proposed to integrate active inference with imitation learning (i.e., AIL) for autonomous vehicles. The presented AIL framework is based on learning a situation model encoded in a coupled Generalized Dynamic Bayesian Network explaining the dynamic interactions between two moving vehicles (i.e., an expert agent and a dynamic object). The situation model is used to initialize a First-Person model, which the learner agent can use to predict expert-object dynamic interactions and evaluate the situation. During the online process, the learner agent is equipped with an Active First-Person model consisting of the First-person model and active states representing actions, thus enriching the learner agent with the capability to predict expert dynamics and expected relative distance from a moving object in order to perform efficient actions. The learner agent relies on an abnormality indicator that measures how much observations support its expectations to decide whether to imitate the expert's behavior under normal situations or explore new actions in abnormal situations (i.e., unseen by the expert). Under the active inference approach, we showed how the learner could learn a new set of configurations and actions incrementally that allow the learner to optimize internal predictions (about the surrounding environment) and action selection (to come near the situation model) jointly, leading to free energy minimization. Experimental results have shown that perceptual learning and inference are required to induce prior expectations about how new experiences and abnormalities unfold. Action is being taken to resample the world in order to meet these expectations. This places perception and action together to drive solely based on the free energy measurement policies and conducts experiments regarding general applicability to autonomous driving and generalization between different changes in dynamic environments. In addition, results have indicated that the proposed approach outperforms reinforcement learning methods such as Q-learning, double Q-learning, and inverse reinforcement learning in terms of the number of selected actions, successful travel rate, collision probability, going out of boundary probability, and imitation loss.

Model II - Employ modified MJPF to Active Inference proposed a hybrid mechanism integrating active inference with imitation learning to enhance autonomous driving skills. Particularly, the Markov Jump Particle Filter is implemented to perform joint predictions of configurations and Generalized States during the online learning phase. The proposed approach allows a learning agent imitates suboptimal driving policy based on a probabilistic situation model (encoded in a coupled Generalized Dynamic Bayesian Network) learned from expert demonstrations, adapt to dynamic changes in the environment (i.e., unseen situations), and perform safe movements without colliding with another dynamic object interacting in

the environment. This section showed how the learner's predictive capability allows deciding whether to exploit the expert's policy during normal situations (experienced by the expert) or to explore new actions and update the dynamic model incrementally during abnormal situations (newly discovered). It demonstrated that the learner's objective is to take the best set of actions during both the exploration and exploitation phases that minimize the free energy (or maximize reward). Experimental results show the effectiveness of the proposed method in imitating optimal expert's driving policy and adapting to unseen situations to accomplish a takeover task. In addition, results have indicated that the proposed approach outperforms other reinforcement learning methods.

Chapter 5

Exploring Action-Oriented Models via Active Inference for Autonomous Vehicles

5.1 Introduction

Autonomous driving systems (ADS) are generally partitioned into a hierarchical structure, including perception, decision making, action planning, and vehicle control [28]. Perception and navigation in a dynamic environment have been a long-standing challenge in AVs. In addition to the complexity of the decision-making systems that might provoke errors causing performance degradation and lead to severe situations (e.g., collisions) [63]. Performing suitable actions according to the dynamic environmental changes around the AV significantly impacts error minimization. Thus, action planning is still a challenging task responsible for safety and efficiency. It should consider the feasibility constraints in a kinematic and dynamic manner based on the information about the perceived environment and the reasonable prediction of the other contributor's behaviors. Moreover, it should be able to generate optimal or semi-optimal maneuvers that provide suitable driving quality, such as exactitude and consistency.

Satisfying the earlier requirements mandates an efficient theory capable of representing causal relationships in the world and providing optimal behavior in highly uncertain environments. In addition, for an AV to reach a high level of autonomy, it must be equipped with SA. Recent progress in signal processing and ML allows an intelligent learning agent to achieve a SA model by observing multi-sensorial data from an accomplished task by an expert agent.

A SA autonomous system constantly deals with continuous and potentially overwhelming signals from the agent's sensors and their interaction with the dynamic surrounding. For learning and adaptation, the IA must transform the sensory inputs into a reliable perception

of the world. One of the ultimate goals of artificial intelligence is to construct autonomous agents capable of human-level performance. Motivated by this, CS has debated how exactly the brain carries out the learning activities. While previous researches propose perception primarily as a bottom-up readout of sensory signals, emerging Bayesian models suggest, instead, that perception is cognitively modulated and might be best viewed as a process of prediction based on an integration of sensory inputs, prior experience, and contextual cues [106, 98].

The brain executes the Bayes rule to perceive the world by continuously generating a top-down cascade of encoded hypotheses about environmental states and processing bottom-top projections of sensory inputs compared with the prior hypothesis's top-down flow [70]. Any mismatch between top-down predictions and bottom-up sensory responses results in prediction errors, prompting the system to refine its hypotheses. Thus, there is a strong link between bottom-up perception and top-down prediction, allowing to continuously update the priors to better predict the subsequent incoming sensory inputs and minimize errors. In this view, experiences are necessary because they assess how good the model is and give a hint to correct future predictions through the computation of prediction errors. As a result, ascending projections do not capture the characteristics of a stimulus but rather how surprised the brain is by it, given the strong link between surprise and model uncertainty [116].

Consequently, AIn has emerged as a novel theory explaining the idea that the brain is essentially a prediction and inference machine that actively attempts to predict, experiment with, and comprehend its surroundings [34, 114, 94]. Perception and action are strongly linked in AIn in order to minimize the FE [52], both coming from the brain's beliefs about the world and being constrained by sensory inputs from the environment [8].

In this section, motivated by the above discussion and previous work [102], we introduce a SA framework empowered by AIn to improve ADS. The proposed framework consists of three main modules: a **multi-modal perception module**, a **global learning module** (world model) and an **active learning module**. Thus, an AV (learning agent) equipped with SA is capable of learning how to self-drive in a dynamic environment while interacting with another moving agent (i.e., vehicle).

The multi-modal perception module allows the AV to perceive the external world as a bundle of exteroceptive and proprioceptive sensations from multiple sensory modalities (e.g., positional information from GPS sensors, images from cameras, point clouds from Lidar, etc.) and to be able to integrate information from different sensory inputs and match them appropriately. In this work, the AV integrates proprioceptive stimuli (i.e., AV's positions) with exteroceptive stimuli (i.e., the relative distance between AV and another object), describing the integration process using Bayesian inference. The AV relies on the global world module

to encode the dynamics of the surrounding environment that is structured in a hierarchical representation. The idea is to use hierarchical representations underlying multisensory integration to explain best how sensory data are caused in multiple modalities.

The global learning module consists of a **situation model (SM)** representing the dynamic driving behaviour of an expert agent interacting with another agent in the environment that is learned from demonstrations and a **First-person model (FP-M)** enabling the third-agent (i.e., AV) in first person view, so that AV can experience a certain driving task from the expert's real perspective. The situation and the First-person models are represented in Coupled Generalized Dynamic Bayesian Networks (C-GDBNs). The former is composed of two GDBNs representing the two agents interacting in the environment where their hidden variables are stochastically coupled (variables are uncorrelated but have coupled means), and each GDBN has its own private observation. Likewise, FP-M represents the stochastic coupling of the interaction between AV and another agent and the AV's behaviour using a C-GDBN.

The active learning module connects the internal models that the AV holds with the decision-making process by enriching the FP-M with active variables representing the set of actions that the AV can perform and so creating the Active First-Person model (AFP-M). This endows the AV with the capability to predict what will happen next in the surroundings and evaluate the environmental situation to understand how it should behave in first person. Hence, the AV can either follow an offline planned task by executing expert-like manoeuvres during normal situations (i.e., situations experienced by the expert) or by planning at run-time and learning incrementally to resolve uncertainty during unexpected situations (i.e., situations not experienced by the expert). To this purpose, we implement a hybrid mechanism by pulling together imitation learning and active inference, inspired by the brain learning procedure that typically integrates the agent's prior knowledge and its actual observations. The AV uses the mismatches between prediction and observations to jointly improve future predictions and actions to minimize future FE (i.e., prediction errors).

The major contributions of this section are summarized as follows:

- It advances a probabilistic computational account of action, observation and imitation abilities grounded in the framework of active inference. While our proposal is domain-general, in this paper, we illustrate it using driving tasks (i.e., lane changing) in a dynamic environment, where a naive learning agent infers and imitates the actions executed by an expert agent.

- The proposed approach enables AV to follow an offline planned task by executing expert-like overtaking manoeuvres in automated driving systems while still taking

autonomous decisions at run-time and learning incrementally to adapt to unexpected situations.

- A probabilistic framework is developed to solve the exploration-exploitation dilemma by foreseeing actions that minimize the prediction errors and establish a solid foundation for further research on the representation and learning of concepts in a cognitive environment by an autonomous agent.
- An online evaluation of joint state predictions is applied to update the belief during the back-projection of detected errors at continuous and discrete levels. We employ a Bayesian sequential decision-making model (i.e., Particle filter, Kalman filter) to distinguish exploration and exploitation processes, which train AV to generate the preferred performance or explore a new course of actions based on its sensory observations and new information provided by the perception of the surrounding.
- Extensive simulations on various overtaking tasks illustrate that the performance of the proposed approach outperforms that of RL. Furthermore, we discuss how clustering of new experiences might affect the performance of the AV in generalizing what has been learned so far to unseen situations.

5.2 System Model - Self-awareness Architecture for Autonomous Driving

The proposed SA architecture depicted in Fig. 5.1 is composed of several modules forming the perception-action cycle that links an AV to its environment. When facing a new situation, an AV makes sense of the external world by creating and testing hypotheses about how the world evolves. It makes predictions based on prior knowledge acquired from past experiences, takes actions based on those hypotheses, perceives the consequences, and adjusts the hypotheses. The different modules in the architecture can be seen as different areas of the biological brain, each one handling particular functionalities. Some parts handle sensory perception, such as seeing, while others handle planning and decision-making. All parts of the brain work together, with messages following between them. As shown in Fig. 5.1, the hierarchical message passing through the levels is not regarded as a straightforward action-feedback mapping. It is represented by inferences and perceptions across different modalities of proprioceptive and exteroceptive sensory signals. Learning this association allows the model to predict the perceptual consequences of acting. Additionally, the model must use these representations to reduce prediction errors and predict how sensory signals change under

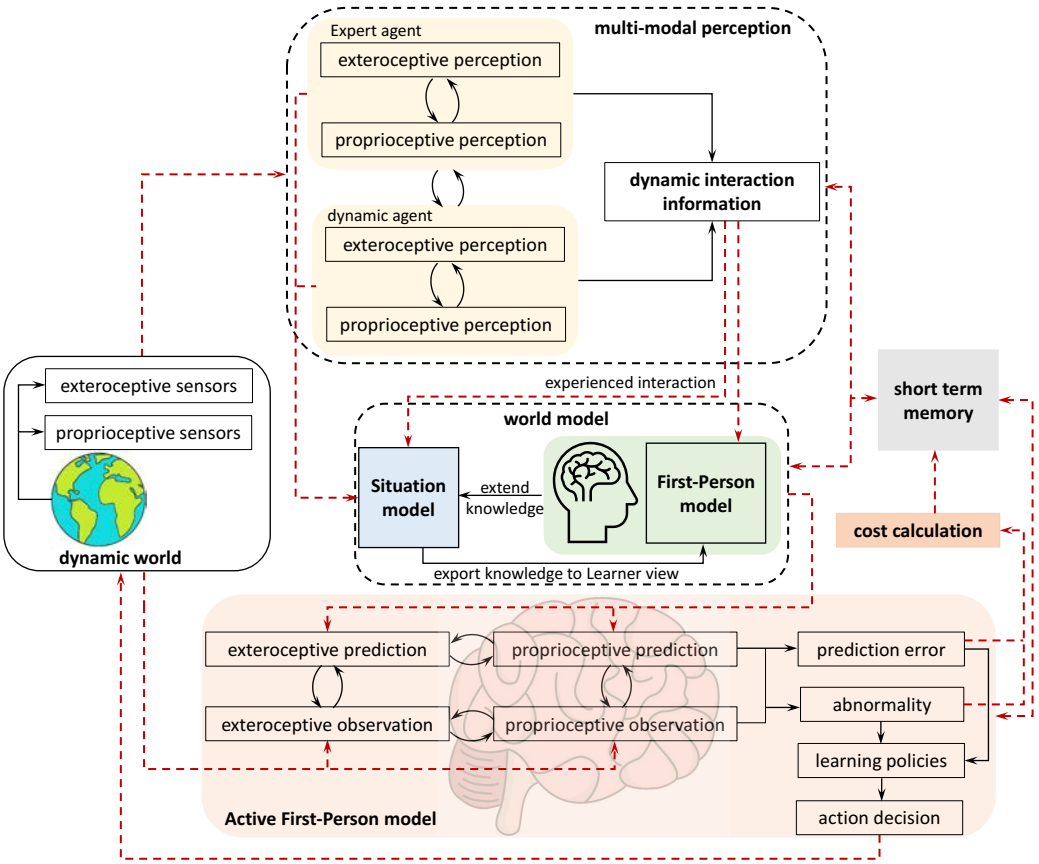


Fig. 5.1 A general schematic of the proposed Self-Awareness architecture for autonomous driving.

specific actions. The following sections present a detailed description of the different modules involved in the architecture.

5.2.1 Dynamic External World

We approach SA from a multi-sensory signal processing perspective in a non-stationary environment. The environment is considered dynamic due to the changing through the agent transitions and simultaneous other processes operating on it. The agent is equipped with exteroceptive sensors to observe the environment and proprioceptive sensors to measure the internal parameters. Accordingly, the agent continuously collects multisensorial data by observing itself and its surroundings and processes the collected data to learn a contextual dynamic representation.

5.2.2 Multi-Modal Perception

A perception system is employed for learning the interaction between an agent and another dynamic object based on multimodal perception using multi-sensorial information. Multimodality enables the model to leverage the presented sensors to identify causalities between multisensory data perceived by the agent. Leveraging multiple sensors to perceive information about the environment is thus crucial when building a model to perform predictions about the agent's dynamics to do motion planning. The perception of multimodal stimuli is an important capability that provides multimodal information in various conditions to enrich the scene library of autonomous driving models.

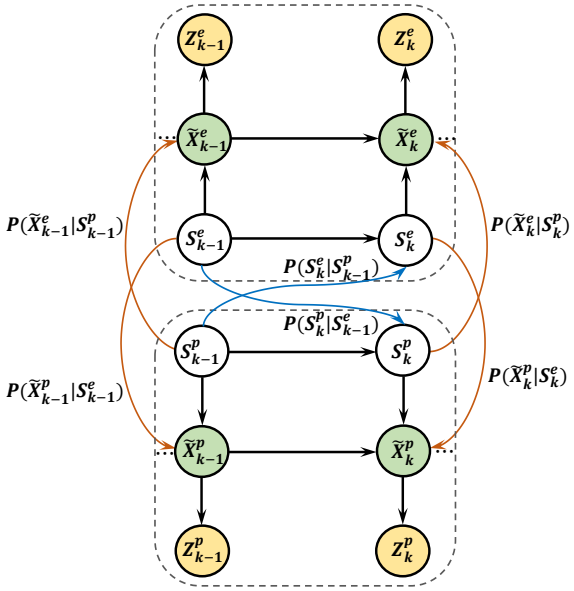


Fig. 5.2 Exteroceptive and proprioceptive information are indexed as e and p, respectively. The orange links describe causalities between both, continuous ($\tilde{X}$) and discrete (S) levels of inference and observed measurements, and the blue links connect both exteroceptive and proprioceptive DBN. This coupling facilitates to model interactions between multisensory data to encode the agent's contextual information.

Fig. 5.2 combines exteroceptive and proprioceptive perception to model a contextual viewpoint for making inferences about future perceived information. Consequently, the context comprises the internal and external perceptions of the agent at each time instant. The main idea is to use such information to predict the following internal or external states. Therefore, the movement of both agent and dynamic object is simulated at each instant by interacting rules that depend on their positions and motions to generate coupled trajectory data. The purpose of analyzing such multisensory data is to encode the coupled agents' dynamic interaction as probabilities into a C-GDBN model. The obtained dynamic interaction model

is self-aware due to its ability to measure the abnormalities and incrementally learn newly interacting behavior derived from an initial one that affects the agent's decision-making.

5.2.3 Global World Model

The world model (WM) plays a simulator role in the brain, and such consideration leads us to take inspiration from the mechanism whereby the brain learns to perform sensorimotor behaviours [79]. In the presented architecture, we obtain the WM using GMs through the interacting experiences from multimodal sensory information. The WM consists of two models, the Situation model (SM) and the First-Person model (FP-M). The SM is an input module that demonstrates the collected sub-optimal information of an expert AV (E) and its interaction with a moving vehicle (O) in a continuous environment where E change-lane frequently and overtake O without an accident. E motion features, O and their interaction are incorporated in a graphical model (i.e., C-GDBN), and the intention of the vehicles can be estimated through probabilistic reasoning. The second model (FP-M) is a transferred generative model. Our focus is attempting to transfer E's knowledge across the First-Person point of view, where an intelligent vehicle (L) learns by interacting with its surroundings via observing the expert behaviour and collecting prior knowledge to incorporate into the environment.

Situation model

The SM is an interactive dynamic model encoding the interactions between two vehicles, namely, E and O, as it is depicted in Fig.5.3. The proposed model assumes synchronized sensory data from both agents' locations. Accordingly, the movement of both agents is simulated at each time instant by interacting rules that depend on their positions and motions. From the E's perspective, it is possible to consider its location measurements as proprioceptive data, whereas the relative position of O represents the exteroceptive information.

The dynamic behaviour of how the two vehicles interact in the environment is described by a generalized hierarchical state-space model in discrete-time comprised of the following equations:

$$\mathrm{D}_k = \mathrm{f}(\mathrm{D}_{k-1}) + \mathrm{w}_k, \tag{5.1a}$$

$$\tilde{\mathrm{X}}_k = \mathrm{g}(\tilde{\mathrm{X}}_{k-1}, \mathrm{D}_k) = \mathrm{F}\tilde{\mathrm{X}}_{k-1} + \mathrm{B}\mathrm{U}_{\mathrm{D}_k} + \mathrm{w}_k, \tag{5.1b}$$

$$\mathrm{Z}_k = \mathrm{h}(\tilde{\mathrm{X}}_k) + \nu_k = \mathrm{H}\tilde{\mathrm{X}}_k + \nu_k. \tag{5.1c}$$

In (5.1a), D_k is a latent discrete state evolving from the previous state D_{k-1} by a non-linear state evolution function $\mathrm{f}(\cdot)$ representing the transition dynamic model and by a Gaussian

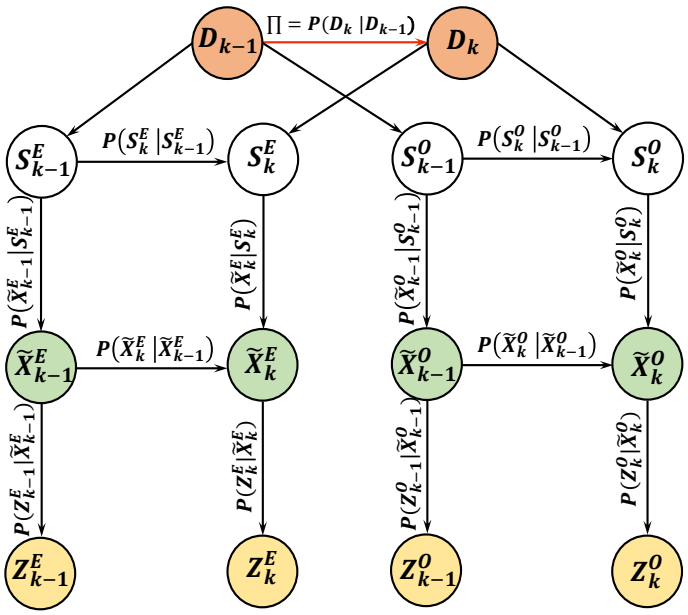


Fig. 5.3 C-GDBN composed of two GDBNs for dynamic interaction. Arrows represent conditional probabilities between the involved variables. Vertical arrows describe causalities between continuous and discrete levels of inference and observed measurements. Horizontal arrows explain temporal causalities between hidden variables. In particular, the red arrow encodes the interaction of a couple of objects.

process noise $\mathrm{w}_k \sim \mathcal{N}(0,\mathrm{Q})$. The discrete state variables $\mathrm{D}_k = [\mathrm{S}^{\mathrm{E}}_k, \mathrm{S}^{\mathrm{O}}_k]$ represent jointly the discrete states of E and O where $\mathrm{S}^{\mathrm{E}}_k \in \mathscr{S}^{\mathrm{E}}$, $\mathrm{S}^{\mathrm{O}}_k \in \mathscr{S}^{\mathrm{O}}$, $\mathrm{D}_k \in \mathscr{D}$, where $\mathscr{S}^{\mathrm{E}}$ and $\mathscr{S}^{\mathrm{O}}$ are learned according to the approach discussed in [102], while $\mathscr{D} = \{\mathrm{D}_1, \mathrm{D}_2, \ldots, \mathrm{D}_m\}$ is the set that represents the dictionary consisting of all the possible joint discrete states (i.e., configurations) and m is the total number of configurations. Observing the configuration's evolution (i.e., joint activated clusters of E and O) over time makes it possible to estimate the transition matrix encoding the probability of switching from one configuration to another, which is defined as:

$$\Pi = \begin{bmatrix} \mathrm{P}(\mathrm{D}_1|\mathrm{D}_1), & \ldots, & \mathrm{P}(\mathrm{D}_1|\mathrm{D}_m) \\ \vdots & \ddots & \vdots \\ \mathrm{P}(\mathrm{D}_m|\mathrm{D}_1), & \ldots, & \mathrm{P}(\mathrm{D}_m|\mathrm{D}_m) \end{bmatrix} \tag{5.2}$$

where $\Pi \in \mathbb{R}^{m,m}$, $\mathrm{P}(\mathrm{D}_i|\mathrm{D}_j)$ represents the transition probability from configuration i to configuration j and $\sum_{k=1}^{m} \mathrm{P}(\mathrm{D}_i|\mathrm{D}_k) = 1 \; \forall i$.

In (5.1b), the continuous latent state $\tilde{\mathrm{X}}_k = [\tilde{\mathrm{X}}^{\mathrm{E}}_k, \tilde{\mathrm{X}}^{\mathrm{O}}_k] \in \mathbb{R}^{n_x}$ represent a joint belief state where $\tilde{\mathrm{X}}^{\mathrm{E}}_k$ and $\tilde{\mathrm{X}}^{\mathrm{O}}_k$ denote the hidden generalized states (GSs) of E and O, respectively. The GSs consist of the vehicles' position and velocity where $\tilde{\mathrm{X}}^i_k = [\mathrm{x}^i_k, \mathrm{y}^i_k, \dot{\mathrm{x}}^i_k, \dot{\mathrm{y}}^i_k]$ and $i \in \{\mathrm{E}, \mathrm{O}\}$. The continuous variables $\tilde{\mathrm{X}}_k$ evolve from the previous state $\tilde{\mathrm{X}}_{k-1}$ by the linear state function

$g(\cdot)$ and by a Gaussian noise w_k. $F \in \mathbb{R}^{n_x,n_x}$ in (5.1b) is the state evolution matrix and $U_{D_k} = \dot{\mu}_{D_k}$ is the control unit vector. In (5.1c), $Z_k \in \mathbb{R}^{n_z}$ is the generalized observation, which is generated from the latent continuous states by a linear function $h(\cdot)$ corrupted by Gaussian noise $v_k \sim \mathcal{N}(0,R)$. Since the observation transformation is linear, there exists the observation matrix $H \in \mathbb{R}^{n_z,n_z}$ mapping hidden continues states to observations.

First-Person Model

FP-M organizes a descriptive dynamic model that enables the third-person (i.e., the learner L) in first-person. So L can experience a driving task from E's real perspective, which facilitates more precise imitative behaviour and allows L to respond quickly and appropriately during the driving task while interacting with another moving vehicle V.

The FP-M is initialized by mapping the hierarchical levels of the SM into FP-M. As shown in Fig. 5.4, the top level of the hierarchy (discrete level) in FP-M represents previously learned configurations ($\mathscr{D}$). So, L through FP-M can regenerate expected interactive manoeuvres that can be used as a reference to evaluate its own interactions with V and infer how the interaction with the external world should be performed.

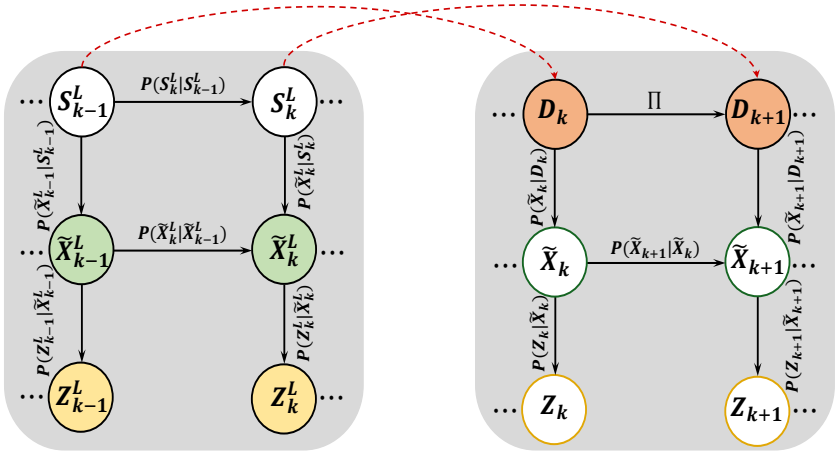


Fig. 5.4 First-Person model. It is composed of an uncoupled proprioceptive model (right side) and the learned joint configuration (left side).

The hidden continuous states in the FP-M represent the dynamic interaction in terms of generalized relative distance consisting of relative distance and relative velocity, which is defined as:

$$X_k = [\tilde{X}^E_k - \tilde{X}^O_k] = [(x^E_k - x^O_k), (\dot{x}^E_k - \dot{x}^O_k)]. \tag{5.3}$$

Likewise, the observations in FP-M depict the measured relative distance between the two vehicles defined as $Z_k = [Z^E_k - Z^O_k]$.

5.3 Online Learning and Inference

Online learning phase provides Active First-Person model (AFP-M). AFP-M connects the WM that L holds with the decision-making block by enriching the FP-M with active states representing the L's actions. Thus, AFP-M represents a generative model $\mathrm{P}(\tilde{\mathrm{Z}},\tilde{\mathrm{X}},\tilde{\mathrm{D}},a)$ of the environment (represented graphically in Fig. 5.5) which is modelled as a partially observed Markov decision process (POMDP).

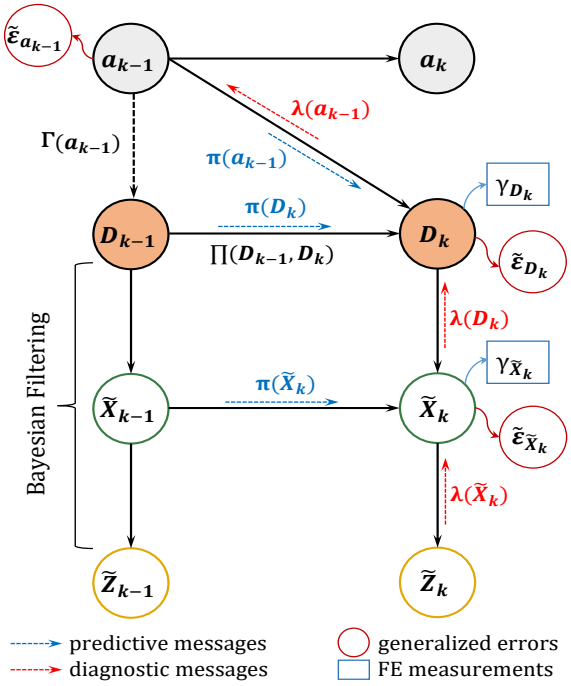


Fig. 5.5 Graphical representation of the Active First-Person model.

AFP-M encompasses joint probability distributions over observations, environmental hidden states at multiple levels and actions performed by L, which is factorized as:

$$\mathrm{P}(\tilde{\mathrm{Z}},\tilde{\mathrm{X}},\tilde{\mathrm{D}},a) = \mathrm{P}(\tilde{\mathrm{D}}_0)\mathrm{P}(\tilde{\mathrm{X}}_0)\prod_{k=2}^{K}\mathrm{P}(\tilde{\mathrm{Z}}_k|\tilde{\mathrm{X}}_k)$$
$$\mathrm{P}(\tilde{\mathrm{X}}_k|\tilde{\mathrm{X}}_{k-1},\tilde{\mathrm{D}}_k)\mathrm{P}(\tilde{\mathrm{D}}_k|\tilde{\mathrm{D}}_{k-1},a_{k-1})\mathrm{P}(a_{k-1}|\tilde{\mathrm{D}}_{k-1}). \quad (5.4)$$

Three hypotheses are considered in a POMPD:

- L does not always have access to the true environmental states but might instead receive observations which are generated according to $\mathrm{P}(\tilde{\mathrm{Z}}_k|\tilde{\mathrm{X}}_k)$ to infer the real states of the environment.

- L operates on beliefs about the hidden environmental states ($\tilde{\mathrm{D}}_k$,$\tilde{\mathrm{X}}_k$) that evolve according to $\mathrm{P}(\tilde{\mathrm{X}}_k|\tilde{\mathrm{X}}_{k-1},\tilde{\mathrm{D}}_k)$ and $\mathrm{P}(\tilde{\mathrm{D}}_k|\tilde{\mathrm{D}}_{k-1},a_{k-1})$.
- L interacts with the external world by seeking to take actions that minimize abnormalities and prediction errors.

5.3.1 Joint Prediction and Perception

Initially (at $k=1$), L relies on prior probability distributions ($\mathrm{P}(\tilde{\mathrm{D}}_0)$, $\mathrm{P}(\tilde{\mathrm{X}}_0)$) to predict the environmental states according to $\tilde{\mathrm{D}}_0 \sim \mathrm{P}(\tilde{\mathrm{D}}_0)$ and $\tilde{\mathrm{X}}_0 \sim \mathrm{P}(\tilde{\mathrm{X}}_0)$, respectively, using a hybrid Bayesian filter called the modified Markov jump particle filter (M-MJPF) [81] consisting of particle filter (PF) and kalman filter (KF). In the successive time instants ($k>1$), L relies on the a priori acquired knowledge of the configurations' evolution given by $\mathrm{P}(\tilde{\mathrm{D}}_k|\tilde{\mathrm{D}}_{k-1})$ which is encoded in (5.2). PF propagates N equally weighted particles drawn from the importance density distribution $\pi(\tilde{\mathrm{D}}_k)=\mathrm{P}(\tilde{\mathrm{D}}_k|\tilde{\mathrm{D}}_{k-1},a_{k-1})$ forming the so-called set of particles $\left\{\tilde{\mathrm{D}}_k^{(i)},w_k^{(i)}\right\}_{i=1}^N$. A bank of KFs is employed for the set of particles to predict the corresponding continuous GSs $\left\{\tilde{\mathrm{X}}_k^{(i)}\right\}_{i=1}^N$ where the prediction of GSs is guided by the upper level as pointed out in (5.1b) that can be expressed in probabilistic form as follows:

$$\mathrm{P}(\tilde{\mathrm{X}}_k^{(i)}|\tilde{\mathrm{X}}_{k-1}^{(i)},\tilde{\mathrm{D}}_k^{(i)}). \tag{5.5}$$

The posterior distribution associated with the predicted GSs is given by:

$$\pi(\tilde{\mathrm{X}}_k^{(i)}) \quad = \quad \mathrm{P}(\tilde{\mathrm{X}}_k^{(i)},\tilde{\mathrm{D}}_k^{(i)}|\tilde{\mathrm{Z}}_{k-1}) \quad = \quad \int \mathrm{P}(\tilde{\mathrm{X}}_k^{(i)}|\tilde{\mathrm{X}}_{k-1}^{(i)},\tilde{\mathrm{D}}_k^{(i)})\lambda(\tilde{\mathrm{X}}_{k-1}^{(i)})\mathrm{d}\tilde{\mathrm{X}}_{k-1}^{(i)}, \tag{5.6}$$

where $\lambda(\tilde{\mathrm{X}}_{k-1}^{(i)})=\mathrm{P}(\tilde{\mathrm{Z}}_{k-1}|\tilde{\mathrm{X}}_{k-1}^{(i)})$ is the diagnostic message propagated previously after observing $\tilde{\mathrm{Z}}_{k-1}$ at time $k-1$. Consequently, once a new observation $\tilde{\mathrm{Z}}_k$ is received, multiple diagnostic messages propagate in a bottom-up manner to update L's belief in hidden environmental states. Thus, updated belief in GSs is given by:

$$\mathrm{P}(\tilde{\mathrm{X}}_k^{(i)},\tilde{\mathrm{D}}_k^{(i)}|\tilde{\mathrm{Z}}_k)=\pi(\tilde{\mathrm{X}}_k^{(i)})\times\lambda(\tilde{\mathrm{X}}_k^{(i)}). \tag{5.7}$$

Whereas belief in discrete hidden states can be updated by updating the particles' weights according to:

$$w_k^{(i)}=w_k^{(i)}\times\lambda(\tilde{\mathrm{D}}_k), \tag{5.8}$$

where $\lambda(\tilde{\mathrm{D}}_k)$ is a discrete probability distribution defined as:

$$\lambda(\tilde{\mathrm{D}}_k) = \left[\frac{\frac{1}{\lambda(\tilde{\mathrm{D}}_k^{(1)})}}{\frac{1}{\sum_{i=1}^{m}\lambda(\tilde{\mathrm{D}}_k^{(i)})}}, \frac{\frac{1}{\lambda(\tilde{\mathrm{D}}_k^{(2)})}}{\frac{1}{\sum_{i=1}^{m}\lambda(\tilde{\mathrm{D}}_k^{(i)})}}, \ldots, \frac{\frac{1}{\lambda(\tilde{\mathrm{D}}_k^{(m)})}}{\frac{1}{\sum_{i=1}^{m}\lambda(\tilde{\mathrm{D}}_k^{(i)})}} \right], \tag{5.9}$$

such that,

$$\lambda(\tilde{\mathrm{D}}_k^{(i)}) = \lambda(\tilde{\mathrm{X}}_k^{(i)})\mathrm{P}(\tilde{\mathrm{X}}_k^{(i)}|\tilde{\mathrm{D}}_k^{(i)}) = \mathscr{D}_B\Big(\lambda(\tilde{\mathrm{X}}_k^{(i)}), \mathrm{P}(\tilde{\mathrm{X}}_k^{(i)}|\tilde{\mathrm{D}}_k^{(i)})\Big) = -\ln \int \sqrt{\lambda(\tilde{\mathrm{X}}_k^{(i)}), \mathrm{P}(\tilde{\mathrm{X}}_k^{(i)}|\tilde{\mathrm{D}}_k^{(i)})}\,\mathrm{d}\tilde{\mathrm{X}}_k^{(i)}, \tag{5.10}$$

where $\mathscr{D}_B$ is the Battacharyya distance and $\mathrm{P}(\tilde{\mathrm{X}}_k|\tilde{\mathrm{D}}_k) \sim \mathscr{N}(\mu_{\tilde{\mathrm{D}}_k}, \Sigma_{\tilde{\mathrm{D}}_k})$.

5.3.2 Learn Action-Oriented Model

L's choice of whether to explore or exploit is guided by its awareness of the interaction with the surrounding environment, which is conditioned directly onto particle beliefs. L uses the updated particles' weights to evaluate the encountered situation among familiar with (i.e., already seen by E) or not familiar with (i.e., a novel situation not seen by E) as illustrated in Fig. 5.6 and Fig. 5.7 respectively.

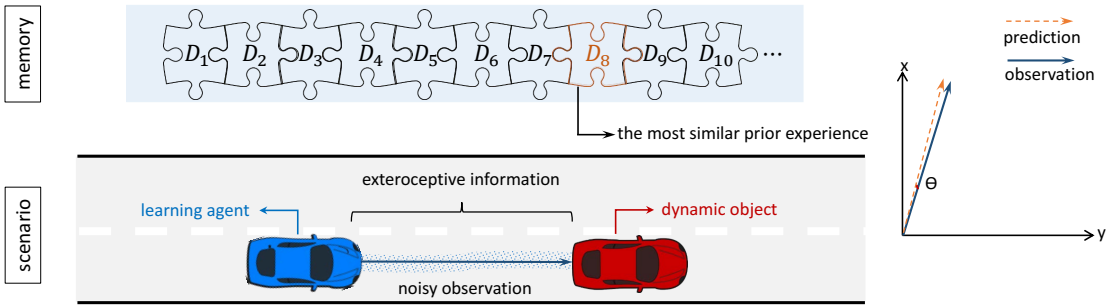


Fig. 5.6 Observing a familiar configuration. The learning agent has proper knowledge about its current interaction with the other dynamic object in the environment.

Thus, L selects an action a_k according to:

$$a_k = \begin{cases} \underset{a_k \in \mathscr{A}}{\operatorname{argmax}}\, \mathrm{P}(a_k|\mathrm{D}_k^{\beta}), \text{ if } \varepsilon_k < \rho \text{ (exploitation)}, \\ \mathrm{q}\big(a_{k-1}, \tilde{\mathscr{E}}_{\tilde{\mathrm{X}}_k^{\beta}}\big), \text{ if } \varepsilon_k \geq \rho \text{ (exploration)}. \end{cases} \tag{5.11}$$

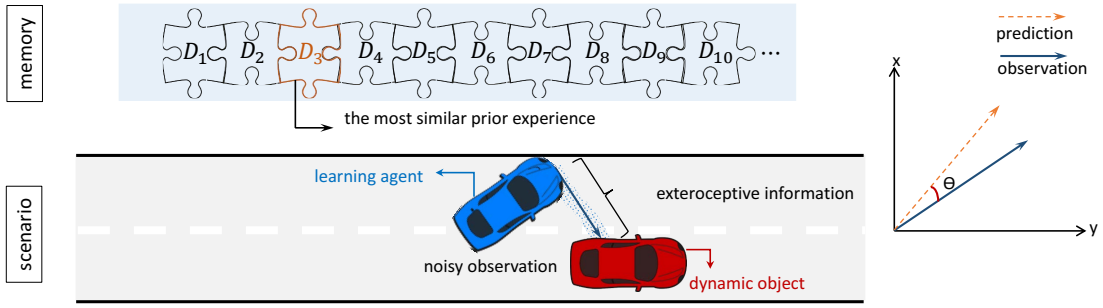


Fig. 5.7 Observing a novel configuration. The learning agent experiences a new interaction with the dynamic object than the learned configurations.

In (5.11), if $\varepsilon_k < \rho$, this means that L is facing similar situation encountered by E and so it will imitate E's action selected from the active inference table Γ defined as:

$$\Gamma = \begin{bmatrix} \mathrm{P}(a_1|\mathrm{D}_1), & \mathrm{P}(a_2|\mathrm{D}_1), & \ldots, & \mathrm{P}(a_m|\mathrm{D}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{P}(a_1|\mathrm{D}_m), & \mathrm{P}(a_2|\mathrm{D}_m), & \ldots, & \mathrm{P}(a_m|\mathrm{D}_m) \end{bmatrix} \tag{5.12}$$

where $\sum_{i=1}^{m} \mathrm{P}(a_i|\mathrm{D}_k) = 1 \ \forall k$, $\mathrm{P}(a_i|\mathrm{D}_j) = \frac{1}{m}$ is the probability of selecting action $a_i \in \mathscr{A}$ conditioned to be in configuration $\mathrm{D}_j \in \mathscr{D}$, $\mathscr{A} = \{\dot{\mu}_{\mathrm{D}_1}, \dot{\mu}_{\mathrm{D}_2}, \ldots, \dot{\mu}_{\mathrm{D}_m}\}$ is the set of available actions, ε_k is the exploration rate given by:

$$\varepsilon_k = 1 - \alpha_k, \tag{5.13}$$

where α_k is the weight of the winning particle computed as:

$$\alpha_k = \max_i \{w_k^{(i)}\}_{i=1}^{N}, \tag{5.14}$$

such that $0 \leq \alpha_k \leq 1$. In addition, β denotes the index of the particle with the maximum weight given by:

$$\beta = \underset{i}{\operatorname{argmax}} \ \{w_k^{(i)}\}_{i=1}^{N}. \tag{5.15}$$

In (5.11), if $\varepsilon_k \geq \rho$, this means that L is facing a novel situation not seen before by E and so L will explore new actions by using the GEs as explained in the coming sections. Fig.5.8 shows a takeover situation example, including explored and exploited trajectories.

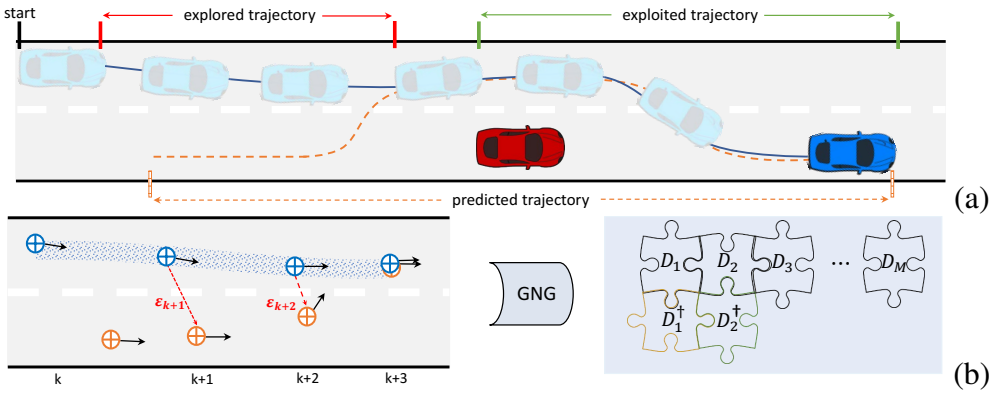


Fig. 5.8 A schematic view of a takeover situation example that is used in our study consists of a) exploratory behavior due to minimizing the divergence with the predicted trajectory and b) associating exploratory clusters to the learning model. L clusters the newly discovered configuration and the novel calculated action using the GNG method.

5.3.3 Abnormality Indicators and Generalized Errors

The predictive messages (i.e., $\pi(\tilde{\mathrm{D}}_k)$, $\pi(\tilde{\mathrm{X}}_k^{(i)})$) propagated top-down the hierarchy are compared against sensory responses signalled via diagnostic messages (i.e., $\lambda(\tilde{\mathrm{X}}_k^{(i)})$, $\lambda(\tilde{\mathrm{D}}_k)$) passing from bottom to up the hierarchy, resulting in multiple abnormality indicators and GEs. Evaluating the abnormality measurement at a certain node allows evaluating to what extent the current observations support the model's predictions, while the GEs allow understanding of how we can suppress those abnormalities in the future. The multi-level abnormality indicators are defined as:

$$\Upsilon_{\tilde{\mathrm{D}}_k} = \mathscr{D}_{\mathscr{KL}}\big(\pi(\tilde{\mathrm{D}}_k), \lambda(\tilde{\mathrm{D}}_k)\big) + \mathscr{D}_{\mathscr{KL}}\big(\lambda(\tilde{\mathrm{D}}_k), \pi(\tilde{\mathrm{D}}_k)\big), \tag{5.16}$$

$$\Upsilon_{\tilde{\mathrm{X}}_k^{(i)}} = -ln\Big(\mathscr{BC}(\pi(\tilde{\mathrm{X}}_k^{(i)}), \lambda(\tilde{\mathrm{X}}_k^{(i)}))\Big), \tag{5.17}$$

where $\mathscr{D}_{KL}$ is the Kullback–Leibler divergence and $\mathscr{BC}$ is the Bhattacharyya coefficient. The GE associated with (5.16) and conditioned on transiting from $\tilde{\mathrm{D}}_{k-1}$ is defined as:

$$\tilde{\mathscr{E}}_{\tilde{\mathrm{D}}_k} = [\tilde{\mathrm{D}}_k, \mathrm{P}(\dot{\mathscr{E}}_{\tilde{\mathrm{D}}_k})] = [\tilde{\mathrm{D}}_k, \lambda(\tilde{\mathrm{D}}_k) - \pi(\tilde{\mathrm{D}}_k)], \tag{5.18}$$

where $\dot{\mathscr{E}}_{\tilde{\mathrm{D}}_k}$ is an aleatory variable described by a discrete probability density function (pdf) $\mathrm{P}(\dot{\mathscr{E}}_{\tilde{\mathrm{D}}_k})$. While the GE projected on the GS space and associated with (5.17) can be expressed as:

$$\tilde{\mathscr{E}}_{\tilde{\mathrm{X}}_k^{(i)}} = [\tilde{\mathrm{X}}_k^{(i)}, \mathrm{P}(\dot{\mathscr{E}}_{\tilde{\mathrm{X}}_k^{(i)}})] = [\tilde{\mathrm{X}}_k^{(i)}, \mathrm{H}^{-1}\tilde{\mathscr{E}}_{\tilde{\mathrm{Z}}_k}], \tag{5.19}$$

where $\dot{\mathscr{E}}_{\tilde{\mathrm{X}}_k^{(i)}}$ is an aleatory variable described by a continuous pdf $\mathrm{P}(\dot{\mathscr{E}}_{\tilde{\mathrm{X}}_k^{(i)}})$ and $\tilde{\mathscr{E}}_{\tilde{\mathrm{Z}}_k} \sim \mathscr{N}(\tilde{\mu}_{\tilde{\mathscr{E}}_{\tilde{\mathrm{Z}}_k}}, \tilde{\Sigma}_{\tilde{\mathscr{E}}_{\tilde{\mathrm{Z}}_k}})$ characterized by the following statistical properties:

$$\tilde{\mu}_{\tilde{\mathscr{E}}_{\tilde{\mathrm{Z}}_k}} = \tilde{\mathrm{Z}}_k - \mathrm{H}\tilde{\mathrm{X}}_k, \tag{5.20}$$

$$\Sigma_{\tilde{\mathscr{E}}_{\tilde{\mathrm{Z}}_k}} = \mathrm{H}\Sigma_{\tilde{\mathscr{E}}_{\tilde{\mathrm{Z}}_k}}\mathrm{H}^{\intercal} + \mathrm{R}, \tag{5.21}$$

where $\tilde{\mu}_{\tilde{\mathscr{E}}_{\tilde{\mathrm{Z}}_k}}$ is the Kalman innovation computed in the measurement space and $\Sigma_{\tilde{\mathscr{E}}_{\tilde{\mathrm{Z}}_k}}$ is the innovation covariance.

5.3.4 Incremental Active Learning and Inference

Active learning and active inference aim to reduce surprises (or abnormalities), either by developing a reliable world model or actively engaging with the environment [103]. Active learning allows an agent to build a predictive model capturing the novel world's regularities through *model parameter exploration*. In contrast, AIn allows using the WM to infer the current context and consequently to infer what to do through the *active states exploration*. These two types of exploration provide a balanced trade-off between adaptive behaviour that aims to minimize abnormalities by fulfilling the learner's preferences on the one hand and acquiring information about the world on the other hand.

Active states exploration: When encountering surprising conditions, L can discover new actions to avoid future abnormal situations. While L is exploring, its new actions evolve from the previous actions and current GEs by a linear function $\mathrm{q}(\cdot)$ as pointed out in (5.11), which is calculated with the first-order Euler integration as follows:

$$\mathrm{q}\big(a_{k-1}, \tilde{\mathscr{E}}_{\tilde{\mathrm{X}}_k^{(\beta)}}\big) = a_{k-1} + \Delta_k \mathrm{P}(\dot{\mathscr{E}}_{\tilde{\mathrm{X}}_k^{(\beta)}}), \tag{5.22}$$

where Δ_k is the step size, a_{k-1} is the previous performed action and $\mathrm{P}(\dot{\mathscr{E}}_{\tilde{\mathrm{X}}_k^{(\beta)}})$ is the GE's pdf defined in (5.19).

Model parameter exploration: Under abnormal conditions and during exploration, L can cluster the novel situations and encode them incrementally in the WM by updating the transition matrix and the active inference matrix, respectively. It is to note that during abnormal situations, new configurations might appear representing novel relative distances between L and the other dynamic object not experienced by E. Thus, clustering the observed relative distance along with the new actions will lead to discovering new configurations and learning how to behave by facing them in the future. Consequently, a set $\mathscr{C}$ consisting of the relative distance-action pair can be performed during the abnormal period T (i.e., during

exploration) as $\mathscr{C} = \{\tilde{\mathrm{Z}}_k, a_k\}_k^T$ which can be used as input to the Growing Neural Gas (GNG) for unsupervised clustering. GNG outputs a set of new configurations defined as:

$$\mathscr{D}^{'} = \{\mathrm{D}_{m+1}, \mathrm{D}_{m+2}, \ldots, \mathrm{D}_{m+n}\} = \{\mathrm{D}_1^{'}, \mathrm{D}_2^{'}, \ldots, \mathrm{D}_n^{'}\}, \tag{5.23}$$

where n is the total number of the newly acquired configurations and $\mathrm{D}_l^{'} \sim \mathscr{N}(\mu_{\mathrm{D}_l^{'}}, \Sigma_{\mathrm{D}_l^{'}})$ such that $\mathrm{D}_l^{'} \in \mathscr{D}^{'}$. Analysing the dynamic evolution of the new configurations allows estimating the transition probability $\mathrm{P}(\tilde{\mathrm{D}}_k|\tilde{\mathrm{D}}_{k-1})$ encoded in $\Pi^{'}$, which is defined as:

$$\Pi^{'} = \begin{bmatrix} \mathrm{P}(\mathrm{D}_1^{'}|\mathrm{D}_1^{'}), & \ldots, & \mathrm{P}(\mathrm{D}_1^{'}|\mathrm{D}_n^{'}) \\ \vdots & \ddots & \vdots \\ \mathrm{P}(\mathrm{D}_n^{'}|\mathrm{D}_1^{'}), & \ldots, & \mathrm{P}(\mathrm{D}_n^{'}|\mathrm{D}_n^{'}) \end{bmatrix}, \tag{5.24}$$

where $\sum_{k=1}^{m} \mathrm{P}(\mathrm{D}_i^{'}|\mathrm{D}_k^{'}) = 1 \ \forall i$. Consequently, the updated global transition matrix $\Pi^{''} \in \mathbb{R}^{(m+n),(m+n)}$ is expressed as:

$$\Pi^{''} = \begin{bmatrix} \Pi & 0_{m,n} \\ 0_{n,m} & \Pi^{'} \end{bmatrix}, \tag{5.25}$$

where Π is the original transition matrix and $\Pi^{'}$ is the newly acquired one.

Likewise, the newly discovered action-configuration pairs characterized by $\mathrm{P}(a_k^{'}|\mathrm{D}_1^{'})$ are encoded in $\Gamma^{''}$ according to:

$$\Gamma^{'} = \begin{bmatrix} \mathrm{P}(a_1^{'}|\mathrm{D}_1^{'}), & \mathrm{P}(a_2^{'}|\mathrm{D}_1^{'}), & \ldots, & \mathrm{P}(a_n^{'}|\mathrm{D}_1^{'}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{P}(a_1^{'}|\mathrm{D}_n^{'}), & \mathrm{P}(a_2^{'}|\mathrm{D}_n^{'}), & \ldots, & \mathrm{P}(a_n^{'}|\mathrm{D}_n^{'}) \end{bmatrix}, \tag{5.26}$$

and the AIn table can be adjusted as follows:

$$\Gamma^{''} = \begin{bmatrix} \Gamma & \frac{1}{n}(\mathrm{J}_{m\times n}) \\ \frac{1}{n}(\mathrm{J}_{n\times m}) & \Gamma^{'} \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \cdots & \gamma_{1n} \\ \vdots & \ddots & \vdots \\ \gamma_{n1} & \cdots & \gamma_{nn} \end{bmatrix} \tag{5.27}$$

where $\mathrm{J}_{m\times n} = [a_{ij}]_{m\times n}$ and $\mathrm{J}_{n\times m} = [b_{ji}]_{n\times m}$ are the unit matrices, such that, $a_{ij} = b_{ji} = 1$ $\forall i, j$. It is to note that $\Gamma^{''}$'s row do not summing 1 due to the addition of the unit matrices and

$\Gamma^{'}$. Thus, normalization is needed, and it can be performed as:

$$\hat{\Gamma}^{''} = \begin{bmatrix} \hat{\gamma}_{11} & \cdots & \hat{\gamma}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{\gamma}_{n1} & \cdots & \hat{\gamma}_{nn} \end{bmatrix}, \tag{5.28}$$

where $\hat{\gamma}_{ij} = \frac{\gamma_{ij}}{\sum_{j=1}^{n} \gamma_{ij}} \; \forall i$.

5.3.5 Action-Oriented Model Update

L relies on the abnormality indicators calculated at time k and defined in (5.16) and (5.17) to evaluate the performed actions at time $k-1$. Under abnormal conditions, L learns how to avoid those abnormalities in the future by seeking information about the surrounding environment and how to engage inside it based on the two types of exploration discussed previously.

In contrast, during exploitation and under abnormal conditions, L updates the existing AIn table and transition matrix using the diagnostic messages $(\lambda(\tilde{\mathrm{D}}_k), \lambda(a_{k-1}))$. The existing transition matrix can be updated using the GE defined in (5.18) as follows:

$$\pi^*(\tilde{\mathrm{D}}_k) = \pi(\tilde{\mathrm{D}}_k) + \mathrm{P}(\dot{\mathscr{E}}_{\tilde{\mathrm{D}}_k}). \tag{5.29}$$

The AIn table Γ can be adjusted according to:

$$\pi^*(a_k) = \pi(a_k) + \mathrm{P}(\dot{\mathscr{E}}_{a_k}), \tag{5.30}$$

where $\pi(a_k) = \mathrm{P}(\cdot|\tilde{\mathrm{D}}_k)$ is a specific row in Γ and $\mathrm{P}(\dot{\mathscr{E}}_{a_k})$ is the GE's pdf related to the active states that can be calculated as [80]:

$$\mathscr{E}_{a_{k-1}} = [a_{k-1}, \mathrm{P}(\dot{\mathscr{E}}_{a_{k-1}})] = [a_{k-1}, \lambda(a_{k-1}) - \pi(a_{k-1})], \tag{5.31}$$

where $\lambda(a_{k-1}) = \lambda(\tilde{\mathrm{D}}_k) \times \mathrm{P}(\tilde{\mathrm{D}}_k|a_{k-1})$.

5.4 Simulation and Performance Evaluation

5.4.1 Experimental Data Set

The expert data are collected during the experiments by considering two AVs interaction, called icab 1 and icab 2 [91], see Fig.4.6. Each AV (i) is equipped with both exteroceptive

and proprioceptive sensors. The employed dataset is gained from the odometry trajectories and control parameters to analyze the interactions between AVs. The sensory modules provide 4-dimensional information, including the positions of AV in (x, y) coordinates, and the control parameters consider the AV's velocity $(\dot{x}, \dot{y})$, as follows:

$$Z_k^i = [x_k^i, y_k^i, \dot{x}_k^i, \dot{y}_k^i] \tag{5.32}$$

During the experiment, icab 2 overtakes from the left side of icab 1 while icab 1 is maneuvering straight.

5.4.2 Offline Learning Phase

In this phase, the SM is provided by employing the collected data from two AVs (i.e., icab 1 and icab 2) interactions as explained in 5.2.3. The shaped SM presents 24 joint clusters that encode the dynamic interaction between the two AVs. Fig. 5.9 shows the generated transition matrix from the AVs trajectories. consequently, FP-M is initialized using 24 learned configurations, including the position data and control parameters (explained in 5.2.3).

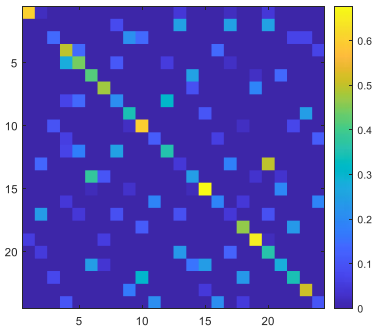


Fig. 5.9 The reference transition matrix based on two AVs movements, icab 1 and icab 2, during the overtaking scenario.

5.4.3 Online Learning Phase

This section evaluates the proposed generalized hierarchical model by considering the following research points:

- Does the model learn accurate representations for inference and prediction by minimizing the generalized errors?
- Can these representations be used in the hierarchical model for generating proper beliefs about the agent's surroundings?

- Does the hierarchical model infer imitative moves and exploratory actions properly to minimize the FE measurements?

The performance of the presented approach in this section is compared with 2 benchmark schemes, AIL, which is proposed in the previous section of this thesis, and the Q-learning algorithm.

5.4.4 Action-Oriented Model

In Fig. 5.10, the learning agent experiences an unobserved trajectory, while trajectory matching via minimizing the prediction error is performed in task space (explained in 5.3.3). Therefore, L uses the exploratory policy (explained in 5.3.5) to solve IL tasks by minimizing the divergence between the observed demonstrations and the expected one. Fig. 5.10 shows the predicted expected trajectory in the red-shaped graph and performed trajectory by the agent is represented by the blue graph. The figure demonstrates that the learned action-oriented model adapts to changing variability in the environment.

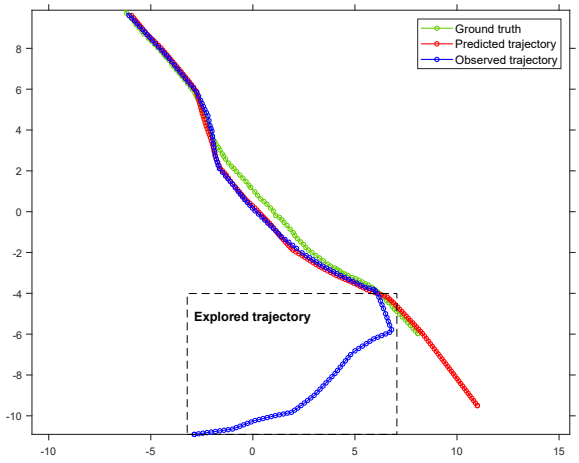


Fig. 5.10 Learner minimizes the distinction between observation and prediction. The action selection procedure is modified by using the detected generalized errors.

Moreover, Fig. 5.11 illustrates the generated clusters and their associated mean actions, which are from the new experiences. Later, L uses the learned model to infer actions from the novel learned configurations. The newly added clusters to the model are demonstrated by yellow circles, which are expanded the original transition model.

Fig. 5.9 shows the original transition matrix (Π) that is provided using the expert behavior composing 24 clusters. During training, the learning agent's Π is modified based on the

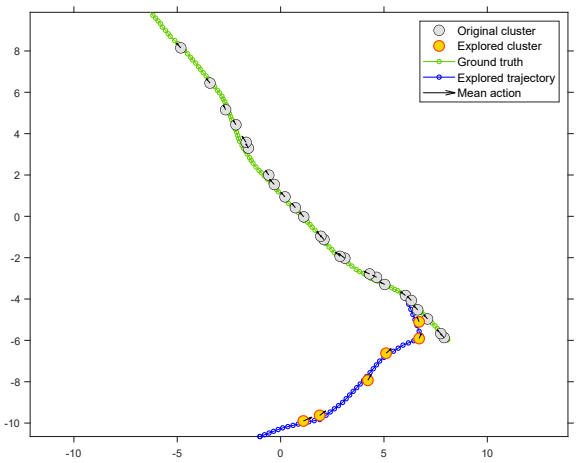
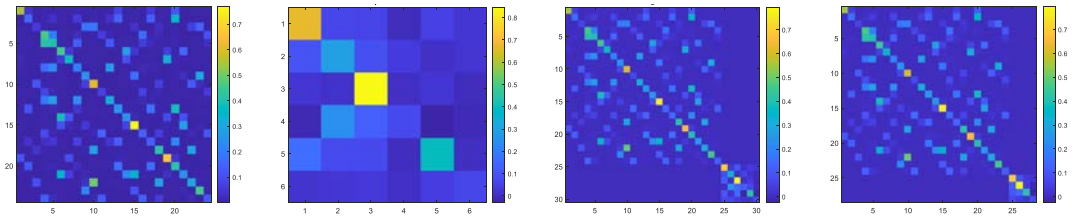


Fig. 5.11 The explored trajectory and the corresponding actions are clustered by GNG.



(a) (b) (c) (d)

Fig. 5.12 Transition matrix evolution during learning new interactions.

learner's movements and decisions when faced with a familiar or novel configuration. The shown transition matrices in Fig 5.12 are based on the considered configurations depicted in Fig. 5.11. Fig 5.12- (a) shows how Π is modified during exploitation from the original clusters (gray clusters in Fig. 5.11), and Fig 5.12- (b) presents the generated transition matrix from the newly explored configurations (yellow clusters in Fig. 5.11). After appending the explored clusters to the main transition model (Fig 5.12- (c)), the learning model refines the learned configuration using a defined threshold to avoid recording mostly similar demonstrations (Fig 5.12- (d)). Table.5.1 explains how the original transition matrix is developed.

Fig.5.13 illustrates the evolution of beliefs about the probabilities of the task for each performed action in the above experiments (see Fig.5.10). Fig.5.13-(a) shows the maximum

Table 5.1 The transition matrix is expanded during learning new observations.

Original Π	Explored Π	Merged Π	Updated Π
24 clusters	6 clusters	30 clusters	28 clusters

particle weight for each experience during 38 configurations (38 performed actions). At the beginning of the experiment, L the distinction between the agent's performance and the expectation is high. Then the assigned weights are low that it leads L to explore and gather experiences that is to learn actively. As Fig.5.13-(b) demonstrates, L makes exploratory and novelty-seeking choices at the start of the experiment. After 15 trials, by modifying the beliefs about the interaction with surroundings, the learning agent could minimize the exploration probability (see Fig.5.13-(c)). Therefore L is confident to behave imitatively which compels it to choose exploitative actions. The presented panel in Fig.5.13-(b) illustrates whether L performs exploratory or exploitative actions as indicated by the blue dots. Darker background implies higher certainty about selecting an exploitative action. Moreover, as Fig.5.13-(d) shows, imitative behavior causes decreasing mean error between the agent's observation and its prediction.

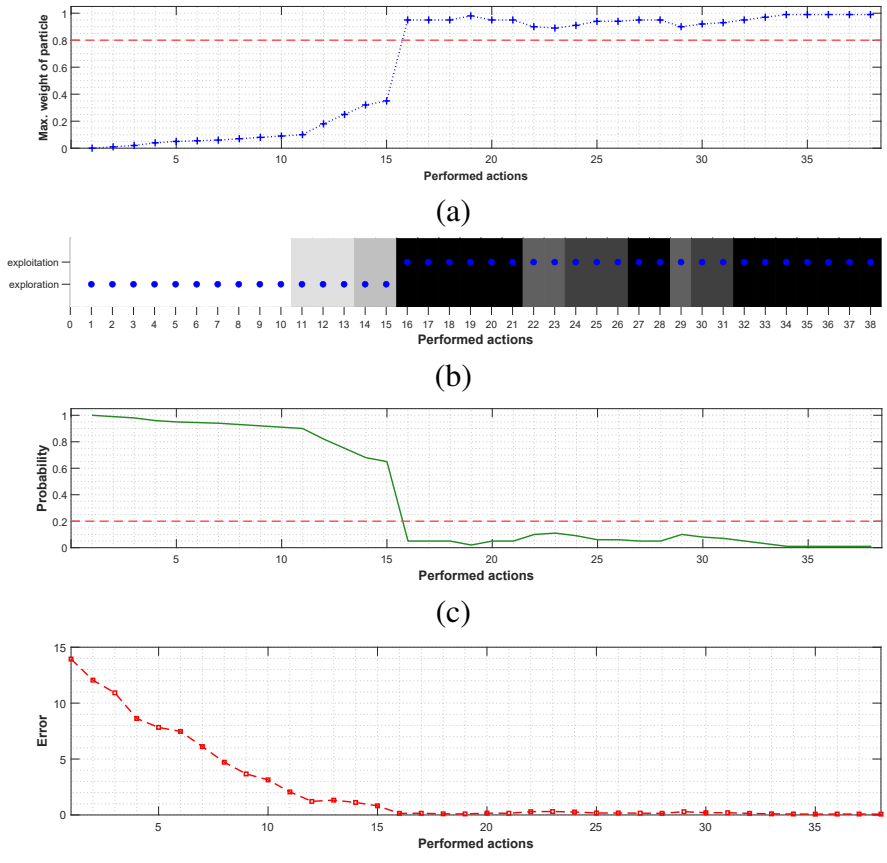


Fig. 5.13 It illustrates belief updates during a simulated experiment by 38 configurations.

The efficiency of the proposed approach is tested in the CARLA simulator [42], which is a three-dimensional (3-D) simulator for AD. CARLA is used in this research to study the performance of the learned model in a 3-D environment similar to the real world. Fig. 5.14 illustrates the AV travel environment. The red rectangles are the states where the experiments are placed in different scenarios as follows:

- The agent is placed in the right lane following the other vehicle in the same lane, which has a slower speed, it needs to overtake from the left side to avoid a collision,
- The agent is placed in the left lane following the other vehicle in the same lane, which has a slower speed, it needs to overtake from the right side to avoid a collision.

Fig. 5.14 CARLA environment. The highlighted space shows the different home lanes of the vehicle.

The results demonstrate the trained agent is able to travel in a different dynamic world properly. Fig.5.15 illustrates some scenes of the experiment where the trained agent (blue vehicle) changes its lane (left side) to avoid collision with the other participant (red vehicle), and after passing the risky interactions, it comes back to its home lane.

Fig.5.16 shows the full path of overtaking from the left side. During overtaking, the trained agent experiences 12 different interactions that must change its movement policy. The panel in Fig.5.17 shows the agent's action selection in each configuration (row). The panel shows the different experiences, and the repeated configurations are erased from it. At each time instant, the agent has 17 action possibilities (learned action from the online learning phase) that each time it exploits the one with the maximum probability (blue cell).

The results prove that the agent learned new interaction (i.e., overtake from the right side) than the expert experiences by exploratory behavior during the online learning phase.

Fig. 5.15 CARLA frames from overtaking from the left side. The blue vehicle is the trained agent that overtakes the red vehicle.

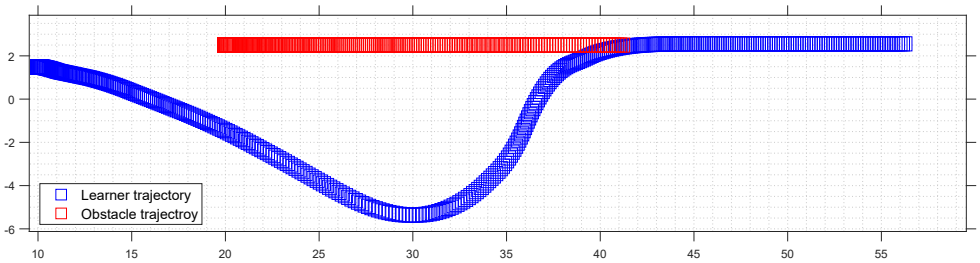


Fig. 5.16 The vehicles' trajectories during overtaking from the left side.

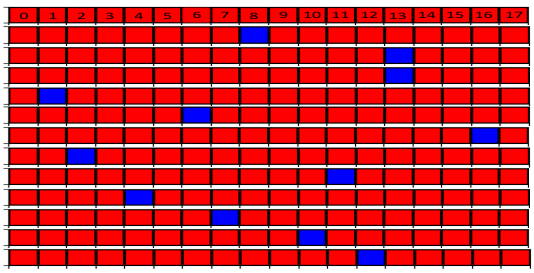


Fig. 5.17 Action panel. Overtaking from the left side.

Fig.5.18 shows the trained agent is able to change its home lane to overtake from the right side of the other vehicle successfully while the collected expert demonstrations were from overtaking from the left side experiences.

Fig.5.19 and Fig.5.20 illustrate the full travel path and action plane during the overtaking from the right side.

Table. 5.2 shows the results of 37 testing travels. Fig.5.21 and Fig.5.22 show an example

Fig. 5.18 CARLA frames from overtaking from the right side. The blue vehicle overtakes the red vehicle.

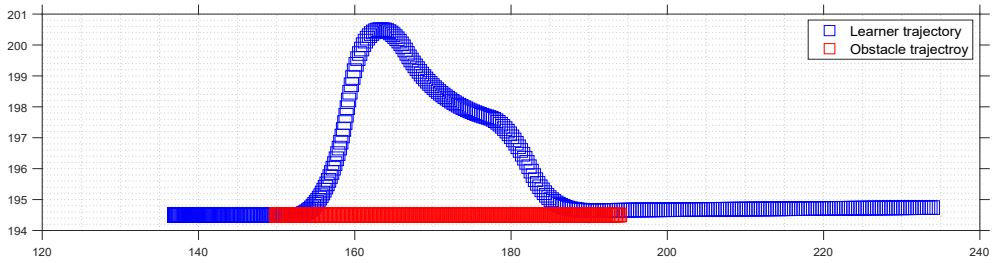


Fig. 5.19 The vehicles' trajectories during overtaking from the right side.

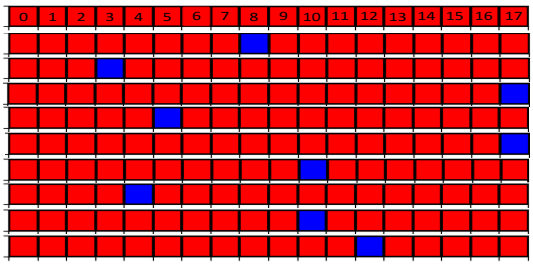


Fig. 5.20 Action panel. Overtaking from the right side.

of failed travel because of a collision with the other vehicle or going out of the street (boundary). Fig.5.23 and Fig.5.24 plot the corresponding full path to Fig.5.21 and Fig.5.22, respectively.

Table 5.2 Results of testing the learned AFP-M in the CARLA simulator.

Success	Loss	Collision	Out of boundary
91.89%	8.10%	2.70%	5.40%
34 travels	3 travels	1 travel	2 travels

Fig. 5.21 CARLA frames from collision experience. The red vehicle collided with the blue vehicle.

Fig. 5.22 Carla Frame from going out of boundary by the agent (blue vehicle).

5.4.5 Cost of Learning

Updating and correcting the beliefs about the agent's surroundings minimize the FE measurement via hierarchical processing in which prior expectations generate top-down predictions of likely observations and where discrepancies between predictions and observations as-

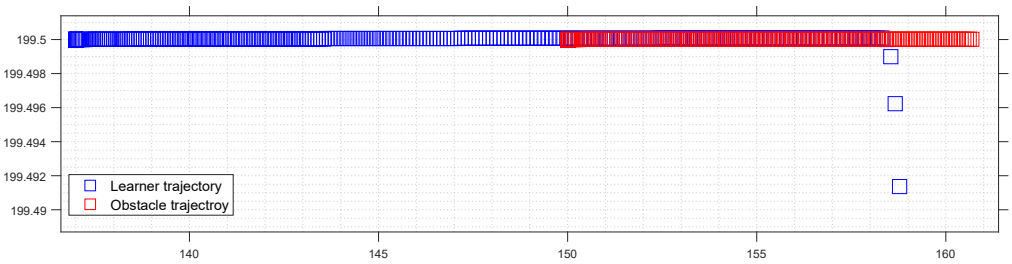


Fig. 5.23 The vehicles' trajectories in the collision case.

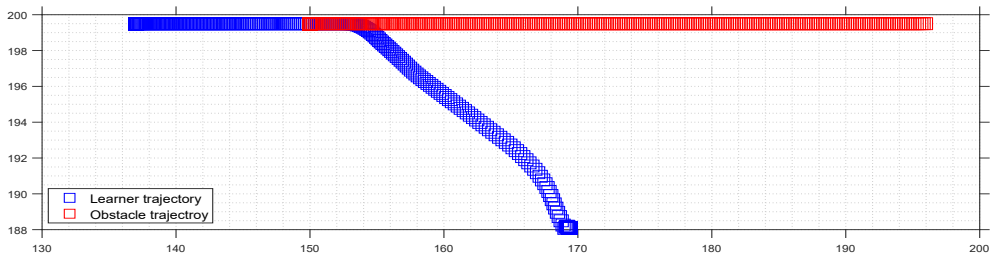
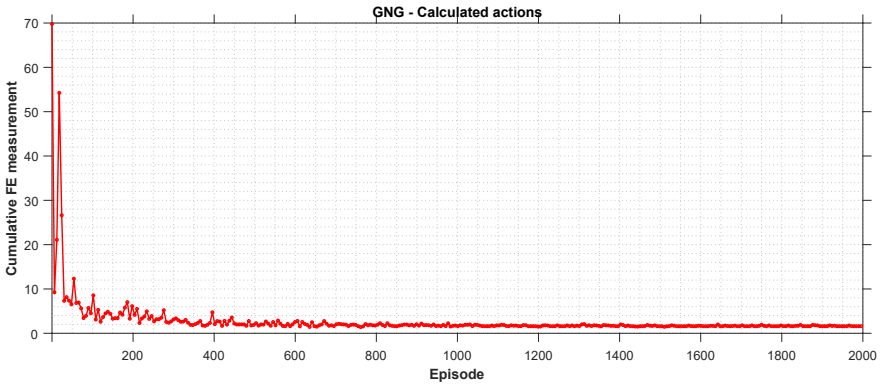


Fig. 5.24 The vehicles' trajectories in going out of boundary case.

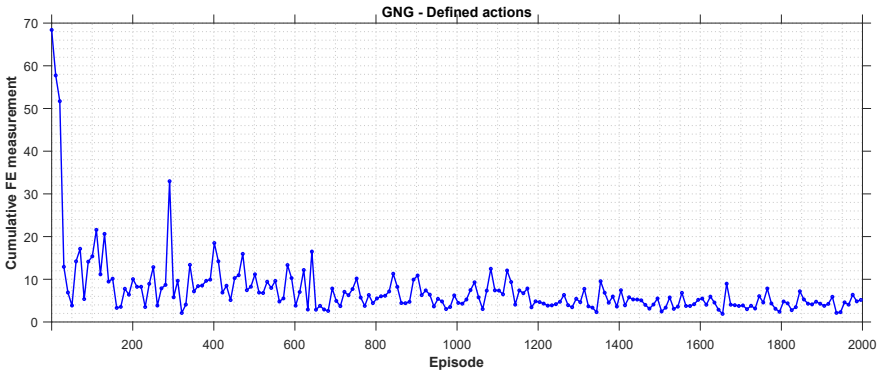
cend to hierarchically higher levels as prediction errors. In this section, the efficiency of 4 action-oriented models is studied in terms of cumulative FE measurement during $2k$ training episodes as follows:

- Model A applies GNG to cluster novel experienced trajectories, and it employs GEs to calculate the exploratory actions,
- Model B applies GNG to cluster novel experienced trajectories, and it uses predefined actions during exploration (discussed in the previous chapter),
- Model C does not cluster the newly observed configurations, and it employs GEs to calculate the exploratory actions,
- Model D does not cluster the newly observed configurations, and it uses predefined actions during exploration.

Fig.5.25-(a) demonstrates the results of Model A, which is the proposed method in this chapter. Comparing Fig.5.25-(a) with the provided results from other models (see Fig.5.25-(b) and Fig.5.26-(a)-(b)) shows clustering the novel configuration and calculating the associated actions using the GEs has a big impact on minimizing the FE measurement during the online learning phase.



(a)



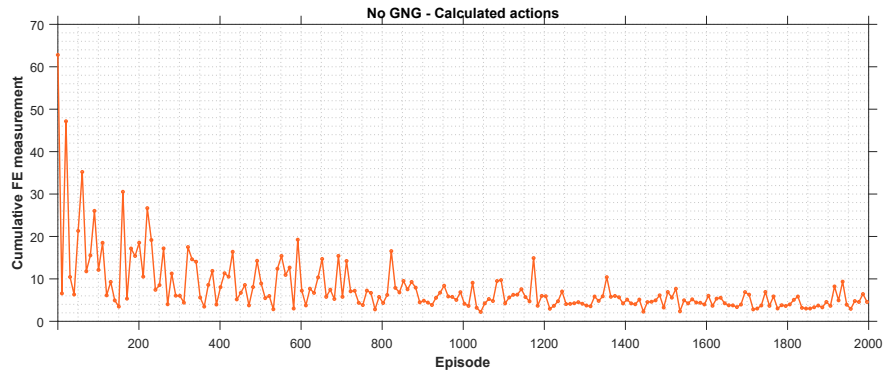
(b)

Fig. 5.25 Calculated cumulative free energy from Model A (a) and Model B (b).

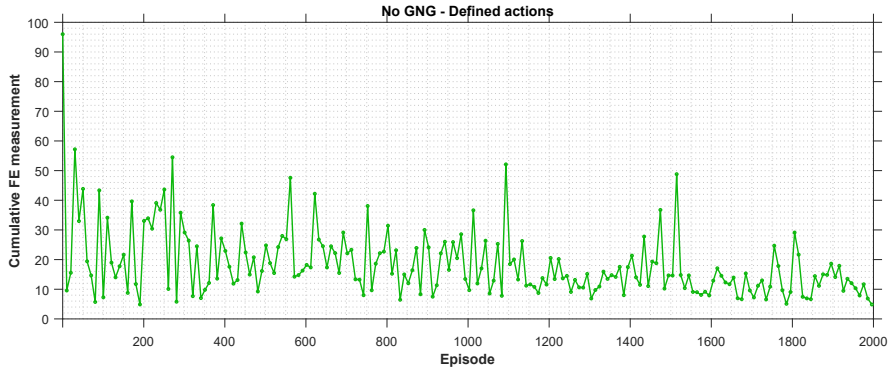
Moreover, Fig.5.27 evaluates the performance of the proposed model in terms of reward (in RL context) than the Q-learning algorithm and AIL, which is introduced in the previous chapter (action-oriented Model B).

5.5 Conclusion

This chapter introduced a hierarchical self-awareness autonomous driving system that advances a probabilistic computational account of action, observation, and imitation abilities grounded in an active inference framework. The autonomous system deals with continuous and potentially overwhelming signals from the vehicle's sensors and their interaction with the dynamic surroundings. For learning and adaptation, the agent must transform the sensory inputs into a reliable perception of the world. The proposed model is composed of several modules forming the perception-action cycle that links the autonomous vehicle to its environment. With inspiration from the biological brain, the different modules are



(a)



(b)

Fig. 5.26 Calculated cumulative free energy from Model C (a) and Model D (b).

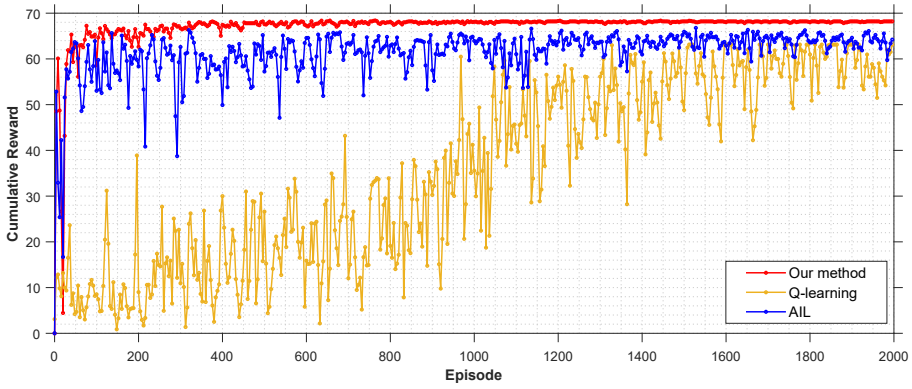


Fig. 5.27 Cumulative reward during $2k$ training episodes

linked together via message passing, where each module handles specific functionalities. The hierarchical message passing is represented by inferences and perceptions across different modalities of multisensorial information (i.e., exteroceptive and proprioceptive data). Learning this association allows the learning model to predict the perceptual consequences

of acting. These representations are employed to minimize prediction errors and accurately predict how sensory signals will change in response to specific actions.

The experimental evaluations show modifying and updating the belief during the online learning phase via the back-projection of detected errors at the multilevel of hierarchy solves the exploration-exploitation dilemma. The autonomous agent generates a preferred movement or an exploratory action based on prior knowledge acquired from past experiences. The hierarchical generalized model learns incrementally from new information provided by the perception of the surroundings to adapt to unexpected situations.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

Artificial intelligence is profoundly changing the world. The autonomous vehicle is a typical complex artificial intelligence system that will become close companions of humans in the future. It is an important and promising direction to explore brain-inspired self-driving technology for the new generation of intelligent transportation systems. Process and understanding multi-sensorial information concerning the cognitive psychological level of the human driving process can significantly improve the cognition ability, decision-making ability, and adaptability to complex situations of self-driving system. A self-driving system based on cognitive construction enables autonomous vehicles to push themselves to higher levels of intelligence through intuitive reasoning and practical learning. Thus autonomous vehicles need a high level of self-awareness to reach full autonomy.

This work shows that self-awareness endows autonomous agents with the capability of maintaining a dynamic equilibrium with the non-stationary world by learning incrementally from new experiences. The agent captures knowledge about itself and structural regularities from its external milieu variations through sensory signals and encoding them in its internal hierarchical Generative models. Furthermore, self-awareness provides various probabilistic inference modes within the Generalized Bayesian Filtering involving predictive top-down messages propagating the belief in hidden variables from high levels of hierarchy towards the lower levels. In turn, bottom-up messages from lower levels report the evidence for expectations of beliefs generating predictions. Comparing predictive (top-down) messages with the sensory responses signaled via diagnostic (bottom-up) messages results in multi-level abnormality indicators and Generalized Errors. Those errors are then fed back from the bottom to up the hierarchy to update beliefs, incrementally encode new concepts, and

finesse plans. Thus, improving future predictions and future actions while minimizing the free energy.

Based on a review of the development of current self-driving technology and the challenges it faces, this study deeply discussed some scientific issues of the self-driving approach based on cognitive construction, as well as the methods, probabilistic computing models, and technical routes to solve these problems. Furthermore, this study expounds on the important role of adapting exploratory behavior mechanisms in realizing robust brain learning in a non-stationary environment. Furthermore, the intuitive reasoning self-driving method based on reinforcement learning, imitation learning, and active inference is also discussed in this study.

Chapter 3 proposed two system models using Generalized Dynamic Bayesian Networks, namely Model I, which is based on a single dynamic agent, and Model II, which considers multi-agent dynamic interaction. Model I proposed an incremental imitation learning model where an intelligent agent tracks a stationary target. The imitator learns the interaction with surroundings by observing an expert agent. This section develops a probabilistic model where the learning agent does not require explicitly repeating the expert agent's behaviors. Therefore, the learner is not limited to recalling exact observations of the optimal behavior but employs a probabilistic model as a flashback memory for guiding a reinforcement learning approach that allows the learning agent to learn the previous experiences on its own. Model II proposed an adaptive probabilistic model for imitation learning in a dynamic environment. In this model, imitation learning is used as a pre-training step to encode the expert demonstrations in a coupled Generalized Dynamic Bayesian Network for reaching a non-stationary target which enables the learning agent to take uncertainty appropriately into account. The presented method demonstrates learning from a dynamic interaction model to minimize the cost of imitation during the online learning phase. In both system models, experimental results show the capability to minimize the abnormalities while learning the policies from the sub-optimal demonstrations. Those abnormalities can be used as qualitative observation in order to learn from unseen situations.

Chapter 4 proposed two system models for autonomous driving in a dynamic environment. In Model I - Active Inference integrated with Imitation Learning, a framework is proposed to integrate active inference with imitation learning (i.e., AIL) for autonomous vehicles. The presented AIL framework is based on learning a situation model encoded in a coupled Generalized Dynamic Bayesian Network explaining the dynamic interactions between two moving vehicles (i.e., an expert agent and a dynamic object). The situation model is used to initialize a First-Person model, which the learner agent can use to predict expert-object dynamic interactions and evaluate the situation. During the online process, the

learner agent is equipped with an Active First-Person model consisting of the First-person model and active states representing actions, thus enriching the learner agent with the capability to predict expert dynamics and expected relative distance from a moving object in order to perform efficient actions. The learner agent relies on an abnormality indicator that measures how much observations support its expectations to decide whether to imitate the expert's behavior under normal situations or explore new actions in abnormal situations (i.e., unseen by the expert). Under the active inference approach, we showed how the learner could learn a new set of configurations and actions incrementally that allow the learner to optimize internal predictions (about the surrounding environment) and action selection (to come near the situation model) jointly, leading to free energy minimization. Experimental results have shown that perceptual learning and inference are required to induce prior expectations about how new experiences and abnormalities unfold. Action is being taken to resample the world in order to meet these expectations. This places perception and action together to drive solely based on the free energy measurement policies and conducts experiments regarding general applicability to autonomous driving and generalization between different changes in dynamic environments. In addition, results have indicated that the proposed approach outperforms reinforcement learning methods such as Q-learning, double Q-learning, and inverse reinforcement learning in terms of the number of selected actions, successful travel rate, collision probability, going out of boundary probability, and imitation loss. Model II - Employ modified MJPF to Active Inference proposed a hybrid mechanism integrating active inference with imitation learning to enhance autonomous driving skills. Particularly, the modified Markov Jump Particle Filter is implemented to perform joint predictions of configurations and Generalized States during the online learning phase. The proposed approach allows a learning agent imitates suboptimal driving policy based on a probabilistic situation model (encoded in a coupled Generalized Dynamic Bayesian Network) learned from expert demonstrations, adapt to dynamic changes in the environment (i.e., unseen situations), and perform safe movements without colliding with another dynamic object interacting in the environment. This section showed how the learner's predictive capability allows deciding whether to exploit the expert's policy during normal situations (experienced by the expert) or to explore new actions and update the dynamic model incrementally during abnormal situations (newly discovered). It demonstrated that the learner's objective is to take the best set of actions during both the exploration and exploitation phases that minimize the free energy (or maximize reward). Experimental results show the effectiveness of the proposed method in imitating optimal expert driving policy and adapting to unseen situations to accomplish a takeover task. In addition, results have indicated that the proposed approach outperforms other reinforcement learning methods.

Chapter 5 proposed a hierarchical self-awareness autonomous driving system that advances a probabilistic computational account of action, observation, and imitation abilities grounded in an active inference framework. The autonomous system deals with continuous and potentially overwhelming signals from the vehicle's sensors and their interaction with the dynamic surroundings. For learning and adaptation, the agent must transform the sensory inputs into a reliable perception of the world. The proposed model is composed of several modules forming the perception-action cycle that links the autonomous vehicle to its environment. With inspiration from the biological brain, the different modules are linked together via message passing, where each module handles specific functionalities. The hierarchical message passing is represented by inferences and perceptions across different modalities of multisensorial information (i.e., exteroceptive and proprioceptive data). Learning this association allows the learning model to predict the perceptual consequences of acting. These representations are employed to minimize prediction errors and accurately predict how sensory signals will change in response to specific actions. The experimental evaluations show modifying and updating the belief during the online learning phase via the back-projection of detected errors at the multilevel hierarchy solves the exploration-exploitation dilemma. The autonomous agent generates a preferred movement or an exploratory action based on prior knowledge acquired from past experiences. The hierarchical generalized model learns incrementally from new information provided by the perception of the surroundings to adapt to unexpected situations.

6.2 Future Works

This thesis has drawn upon concepts from self-awareness, imitation learning, dynamic interaction models, and active inference in autonomous driving systems. Each of these subjects contains a set of existing and emerging methods. Many opportunities exist for extending the ideas presented in this work. Some areas of further study are highlighted below.

- **Develop sensorial input**

  Additional contextual data must be collected to provide compelling and plausible predictions in a real urban environment. Information about the traffic lights and road signs is essential for inferring possible future maneuvers. These cues will be investigated and will create convenient representations in future work. An interesting perspective would be to employ other driving-related sensorial information, such as optical flow or images, as well as their combination.

- **Scenario generation**

  Adopt the obtained error model from interactions between participants for coping with a new driving activity. Adaptability can involve recognizing that a policy or strategy learned in one context can be applied to other situations.

- **Multi-agent generative learning**

  Provide an incremental learning model based on parameter sharing to learn from demonstrations experienced by multiple learning agents. The gathered knowledge from multiple demonstration trajectories will batch together to inform the learning participants.

- **Interact with a third agent**

  Consider unpredictable obstacles (i.e., other vehicles in the urban environment) during driving scenarios, such as changing the home lane in a crowded environment.

References

[1] Airoldi, E. M. (2007). Getting started in probabilistic graphical models. *PLoS Computational Biology*, 3(12):e252.

[2] Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483.

[3] Atkeson, C. G. and Schaal, S. (1997). Robot learning from demonstration. In *ICML*, volume 97, pages 12–20.

[4] Attias, H. (2003). Planning by probabilistic inference. In *International Workshop on Artificial Intelligence and Statistics*, pages 9–16. PMLR.

[5] Bajgar, J., Ciarrochi, J., Lane, R., and Deane, F. P. (2005). Development of the levels of emotional awareness scale for children (leas-c). *British Journal of Developmental Psychology*, 23(4):569–586.

[6] Baker, S. (1897). The identification of the self. *Psychological Review*, 4(3):272.

[7] Baltieri, M. and Buckley, C. L. (2017). An active inference implementation of phototaxis. *arXiv preprint arXiv:1707.01806*.

[8] Barrett, L. F. and Simmons, W. K. (2015). Interoceptive predictions in the brain. *Nature reviews neuroscience*, 16(7):419–429.

[9] Baydoun, M., Campo, D., Kanapram, D., Marcenaro, L., and Regazzoni, C. S. (2019). Prediction of multi-target dynamics using discrete descriptors: an interactive approach. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3342–3346. IEEE.

[10] Baydoun, M., Campo, D., Sanguineti, V., Marcenaro, L., Cavallaro, A., and Regazzoni, C. (2018). Learning switching models for abnormality detection for autonomous driving. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 2606–2613. IEEE.

[11] Behringer, R. and Maurer, R. (1996). Results on visual road recognition for road vehicle guidance. In *Proceedings of Conference on Intelligent Vehicles*, pages 415–420. IEEE.

[12] Bellman, R. (1957). A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684.

[13] Berger-Tal, O., Nathan, J., Meron, E., and Saltz, D. (2014). The exploration-exploitation dilemma: a multidisciplinary framework. *PloS one*, 9(4):e95693.

[14] Bertozzi, M., Broggi, A., Conte, G., Fascioli, A., and Fascioli, R. (1998). Vision-based automated vehicle guidance: the experience of the argo vehicle. *Tecniche di Intelligenza Artificiale e Pattern Recognition per la Visione Artificiale*, pages 35–40.

[15] Bertozzi, M., Broggi, A., and Fascioli, A. (2000). Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous systems*, 32(1):1–16.

[16] Besse, P. C., Guillouet, B., Loubes, J.-M., and Royer, F. (2016). Review and perspective for distance-based clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 17(11):3306–3317.

[17] Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Robot programming by demonstration. In *Springer handbook of robotics*, pages 1371–1394. Springer.

[18] Bioucas-Dias, J. M., Plaza, A., Dobigeon, N., Parente, M., Du, Q., Gader, P., and Chanussot, J. (2012). Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE journal of selected topics in applied earth observations and remote sensing*, 5(2):354–379.

[19] Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

[20] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

[21] Botvinick, M. and Toussaint, M. (2012). Planning as inference. *Trends in cognitive sciences*, 16(10):485–488.

[22] Botvinick, M. and Weinstein, A. (2014). Model-based hierarchical reinforcement learning and human action control. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1655):20130480.

[23] Broggi, A., Bertozzi, M., and Fascioli, A. (2000). Architectural issues on vision-based automatic vehicle guidance: the experience of the argo project. *Real-Time Imaging*, 6(4):313–324.

[24] Broggi, A., Bombini, L., Cattani, S., Cerri, P., and Fedriga, R. I. (2010). Sensing requirements for a 13,000 km intercontinental autonomous drive. In *2010 IEEE Intelligent Vehicles Symposium*, pages 500–505. IEEE.

[25] Broggi, A., Cerri, P., Debattisti, S., Laghi, M. C., Medici, P., Molinari, D., Panciroli, M., and Prioletti, A. (2015). Proud—public road urban driverless-car test. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3508–3519.

[26] Bruineberg, J., Rietveld, E., Parr, T., van Maanen, L., and Friston, K. J. (2018). Free-energy minimization in joint agent-environment systems: A niche construction perspective. *Journal of theoretical biology*, 455:161–178.

[27] Buehler, M., Iagnemma, K., and Singh, S. (2007). *The 2005 DARPA grand challenge: the great robot race*, volume 36. Springer.

[28] Buehler, M., Iagnemma, K., and Singh, S. (2009). *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Springer Publishing Company, Incorporated, 1st edition.

[29] Campo, D., Betancourt, A., Marcenaro, L., and Regazzoni, C. (2017). Static force field representation of environments based on agents' nonlinear motions. *EURASIP Journal on Advances in Signal Processing*, 2017(1):13.

[30] Cerri, P., Soprani, G., Zani, P., Choi, J., Lee, J., Kim, D., Yi, K., and Broggi, A. (2011). Computer vision at the hyundai autonomous challenge. In *2011 14th international IEEE conference on intelligent transportation systems (ITSC)*, pages 777–783. IEEE.

[31] Chalup, S. K. (2002). Incremental learning in biological and machine learning systems. *International Journal of Neural Systems*, 12(06):447–465.

[32] Chen, Z. et al. (2003). Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69.

[33] Chen, Z. and Huang, X. (2017). End-to-end learning for lane keeping of self-driving cars. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1856–1860. IEEE.

[34] Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204.

[35] Colombo, M. and Wright, C. (2021). First principles in the life sciences: the free-energy principle, organicism, and mechanism. *Synthese*, 198(14):3463–3488.

[36] Da Costa, L., Parr, T., Sajid, N., Veselic, S., Neacsu, V., and Friston, K. (2020). Active inference on discrete state-spaces: A synthesis. *Journal of Mathematical Psychology*, 99:102447.

[37] Dayan, P. and Berridge, K. C. (2014). Model-based and model-free pavlovian reward learning: revaluation, revision, and revelation. *Cognitive, Affective, & Behavioral Neuroscience*, 14(2):473–492.

[38] De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D. L. (2000). The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18.

[39] Demiris, Y. (2007). Prediction of intent in robotics and multi-agent systems. *Cognitive processing*, 8(3):151–158.

[40] Diaz, M., Fevens, T., and Paull, L. (2021). Uncertainty-Aware Policy Sampling and Mixing for Safe Interactive Imitation Learning. In *2021 18th Conference on Robots and Vision (CRV)*, pages 72–78.

[41] Doll, B. B., Simon, D. A., and Daw, N. D. (2012). The ubiquity of model-based reinforcement learning. *Current opinion in neurobiology*, 22(6):1075–1081.

[42] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR.

[43] Droniou, A., Ivaldi, S., and Sigaud, O. (2014). Learning a repertoire of actions with deep neural networks. In *4th International Conference on Development and Learning and on Epigenetic Robotics*, pages 229–234. IEEE.

[44] Dutt, N., Regazzoni, C. S., Rinner, B., and Yao, X. (2020). Self-awareness for autonomous systems. *Proceedings of the IEEE*, 108(7):971–975.

[45] Englund, C., Chen, L., Ploeg, J., Semsar-Kazerooni, E., Voronov, A., Bengtsson, H. H., and Didoff, J. (2016). The grand cooperative driving challenge 2016: boosting the introduction of cooperative automated vehicles. *IEEE Wireless Communications*, 23(4):146–152.

[46] Fernando, T., Denman, S., Sridharan, S., and Fookes, C. (2020). Deep inverse reinforcement learning for behavior prediction in autonomous driving: Accurate forecasts of vehicle motion. *IEEE Signal Processing Magazine*, 38(1):87–96.

[47] Friston, K. (2009). The free-energy principle: a rough guide to the brain? *Trends in Cognitive Sciences*, 13(7):293–301.

[48] Friston, K. (2012). The history of the future of the bayesian brain. *NeuroImage*, 62(2):1230–1233.

[49] Friston, K. (2013). Life as we know it. *Journal of the Royal Society Interface*, 10(86):20130475.

[50] Friston, K., Da Costa, L., Hafner, D., Hesp, C., and Parr, T. (2021). Sophisticated inference. *Neural Computation*, 33(3):713–763.

[51] Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., and Pezzulo, G. (2017a). Active inference: a process theory. *Neural computation*, 29(1):1–49.

[52] Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., Pezzulo, G., et al. (2016). Active inference and learning. *Neuroscience & Biobehavioral Reviews*, 68:862–879.

[53] Friston, K., Kilner, J., and Harrison, L. (2006). A free energy principle for the brain. *Journal of physiology-Paris*, 100(1-3):70–87.

[54] Friston, K., Rigoli, F., Ognibene, D., Mathys, C., Fitzgerald, T., and Pezzulo, G. (2015). Active inference and epistemic value. *Cognitive neuroscience*, 6(4):187–214.

[55] Friston, K., Schwartenbeck, P., FitzGerald, T., Moutoussis, M., Behrens, T., and Dolan, R. J. (2014a). The anatomy of choice: dopamine and decision-making. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1655):20130481.

[56] Friston, K., Sengupta, B., and Auletta, G. (2014b). Cognitive dynamics: From attractors to active inference. *Proceedings of the IEEE*, 102(4):427–445.

[57] Friston, K. J., Daunizeau, J., and Kiebel, S. J. (2009). Reinforcement learning or active inference? *PloS one*, 4(7):e6421.

[58] Friston, K. J., Daunizeau, J., Kilner, J., and Kiebel, S. J. (2010). Action and behavior: a free-energy formulation. *Biological cybernetics*, 102(3):227–260.

[59] Friston, K. J., Lin, M., Frith, C. D., Pezzulo, G., Hobson, J. A., and Ondobaka, S. (2017b). Active inference, curiosity and insight. *Neural computation*, 29(10):2633–2683.

[60] Friston, K. J., Rosch, R., Parr, T., Price, C., and Bowman, H. (2018). Deep temporal models and active inference. *Neuroscience & Biobehavioral Reviews*, 90:486–501.

[61] Gangwani, T. and Peng, J. (2020). State-only imitation with transition dynamics mismatch. *arXiv preprint arXiv:2002.11879*.

[62] Gao, X., Zhang, Z.-Y., and Duan, L.-M. (2018). A quantum machine learning algorithm based on generative models. *Science advances*, 4(12):eaat9004.

[63] Garcia, J., Feng, Y., Shen, J., Almanee, S., Xia, Y., and Chen, Q. A. (2020). A Comprehensive Study of Autonomous Vehicle Bugs. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 385–396.

[64] Ghahramani, Z. (1997). Learning dynamic bayesian networks. *International School on Neural Networks, Initiated by IIASS and EMFCSC*, pages 168–197.

[65] González, D., Pérez, J., Milanés, V., and Nashashibi, F. (2015). A review of motion planning techniques for automated vehicles. *IEEE Transactions on intelligent transportation systems*, 17(4):1135–1145.

[66] Hahn, J. and Zoubir, A. M. (2017). Bayesian nonparametric unmixing of hyperspectral images. *arXiv preprint arXiv:1702.08007*.

[67] Hasselt, H. (2010). Double q-learning. *Advances in neural information processing systems*, 23.

[68] Henriksen, M. (2020). Variational free energy and economics optimizing with biases and bounded rationality. *Frontiers in Psychology*, 11:549187.

[69] Hill, D. J., Minsker, B. S., and Amir, E. (2007). Real-time bayesian anomaly detection for environmental sensor data. In *Proceedings of the Congress-International Association for Hydraulic Research*, volume 32, page 503. Citeseer.

[70] Hohwy, J. (2013). *The predictive mind*. OUP Oxford.

[71] Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35.

[72] Inagaki, T. and Sheridan, T. B. (2019). A critique of the sae conditional driving automation definition, and analyses of options for improvement. *Cognition, technology & work*, 21(4):569–578.

[73] Iqbal, H., Campo, D., Baydoun, M., Marcenaro, L., Gomez, D. M., and Regazzoni, C. (2019). Clustering optimization for abnormality detection in semi-autonomous systems. In *1st international workshop on multimodal understanding and learning for embodied applications*, pages 33–41.

[74] Iqbal, H., Campo, D., Marcenaro, L., Gomez, D. M., and Regazzoni, C. (2021). Data-driven transition matrix estimation in probabilistic learning models for autonomous driving. *Signal Processing*, 188:108170.

[75] Jordan, M. I. (2004). Graphical models. *Statistical science*, 19(1):140–155.

[76] Kailath, T. (1967). The divergence and bhattacharyya distance measures in signal selection. *IEEE transactions on communication technology*, 15(1):52–60.

[77] Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory.

[78] Karagiannis, G., Altintas, O., Ekici, E., Heijenk, G., Jarupan, B., Lin, K., and Weil, T. (2011). Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE communications surveys & tutorials*, 13(4):584–616.

[79] Kim, S., Laschi, C., and Trimmer, B. (2013). Soft robotics: a bioinspired evolution in robotics. *Trends in biotechnology*, 31(5):287–294.

[80] Krayani, A., Alam, A. S., Marcenaro, L., Nallanathan, A., and Regazzoni, C. (2022). A Novel Resource Allocation for Anti-Jamming in Cognitive-UAVs: An Active Inference Approach. *IEEE Communications Letters*, 26(10):2272–2276.

[81] Krayani, A., Baydoun, M., Marcenaro, L., Alam, A. S., and Regazzoni, C. (2020). Self-Learning Bayesian Generative Models for Jammer Detection in Cognitive-UAV-Radios. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–7.

[82] Kuefler, A., Morton, J., Wheeler, T., and Kochenderfer, M. (2017). Imitating driver behavior with generative adversarial networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 204–211. IEEE.

[83] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

[84] Lewis, P. R., Platzner, M., Rinner, B., Tørresen, J., and Yao, X. (2016). Self-aware computing: Introduction and motivation. In *Self-aware Computing Systems*, pages 1–5. Springer.

[85] Li, S., Li, N., Girard, A., and Kolmanovsky, I. (2019). Decision making in dynamic and interactive environments based on cognitive hierarchy theory, bayesian inference, and predictive control. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 2181–2187. IEEE.

[86] Lin, Y., Wang, P., and Ma, M. (2017). Intelligent transportation system (its): Concept, challenge and opportunity. In *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*, pages 167–172. IEEE.

[87] Liu, M., Buntine, W., and Haffari, G. (2018). Learning how to actively learn: A deep imitation learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1874–1883.

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

[89] Lungarella, M. and Sporns, O. (2005). Information self-structuring: Key principle for learning and development. In *Proceedings. The 4th International Conference on Development and Learning, 2005*, pages 25–30. IEEE.

[90] Machin, M., Sanguesa, J. A., Garrido, P., and Martinez, F. J. (2018). On the use of artificial intelligence techniques in intelligent transportation systems. In *2018 IEEE wireless communications and networking conference workshops (WCNCW)*, pages 332–337. IEEE.

[91] Marín-Plaza., P. et al. (2016). Stereo Vision-based Local Occupancy Grid Map for Autonomous Navigation in ROS. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: VISAPP, (VISIGRAPP 2016)*, pages 701–706. INSTICC, SciTePress.

[92] Maurer, M., Behringer, R., Furst, S., Thomanek, F., and Dickmanns, E. D. (1996). A compact vision system for road vehicle guidance. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 3, pages 313–317. IEEE.

[93] McLachlan, G. J. (1999). Mahalanobis distance. *Resonance*, 4(6):20–26.

[94] Millidge, B. (2019). Combining active inference and hierarchical predictive coding: A tutorial introduction and case study.

[95] Millidge, B. (2021). Applications of the free energy principle to machine learning and neuroscience. *arXiv preprint arXiv:2107.00140*.

[96] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

[97] Mozaffari, S., Al-Jarrah, O. Y., Dianati, M., Jennings, P., and Mouzakitis, A. (2020). Deep learning-based vehicle behavior prediction for autonomous driving applications: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):33–47.

[98] Munro, D. (2021). Perceiving as knowing in the predictive mind. *Philosophical Studies*, pages 1–27.

[99] Murphy, K. P. (2002). *Dynamic bayesian networks: representation, inference and learning*. University of California, Berkeley.

[100] Nanni, M. and Pedreschi, D. (2006). Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, 27(3):267–289.

[101] Ng, A. Y., Russell, S., et al. (2000). Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2.

[102] Nozari, S., Krayani, A., Marin-Plaza, P., Marcenaro, L., Gómez, D. M., and Regazzoni, C. (2022). Active Inference Integrated With Imitation Learning for Autonomous Driving. *IEEE Access*, 10:49738–49756.

[103] Ofner, A. and Stober, S. (2020). Balancing Active Inference and Active Learning with Deep Variational Predictive Coding for EEG. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3839–3844.

[104] Ogishima, R., Karino, I., and Kuniyoshi, Y. (2020). Combining imitation and reinforcement learning with free energy principle.

[105] Ondruš, J., Kolla, E., Vertal', P., and Šarić, Ž. (2020). How do autonomous cars work? *Transportation Research Procedia*, 44:226–233.

[106] Ongaro, G. and Kaptchuk, T. J. (2019). Symptom perception, placebo effects, and the bayesian brain. *Pain*, 160(1):1.

[107] Onishi, T., Motoyoshi, T., Suga, Y., Mori, H., and Ogata, T. (2019). End-to-end learning method for self-driving cars with trajectory recovery using a path-following function. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[108] Orlande, H., Colaço, M., Dulikravich, G., Vianna, F., da Silva, W., da Fonseca, H., and Fudym, O. (2011). Tutorial 10 kalman and particle filters. *Advanced Spring School: Thermal measurements and inverse techniques*, 5:1–39.

[109] Paden, B., Čáp, M., Yong, S. Z., Yershov, D., and Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55.

[110] Pakdamanian, E., Sheng, S., Baee, S., Heo, S., Kraus, S., and Feng, L. (2021). Deeptake: Prediction of driver takeover behavior using multimodal data. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

[111] Papathanasopoulou, V. and Antoniou, C. (2015). Towards data-driven car-following models. *Transportation Research Part C: Emerging Technologies*, 55:496–509.

[112] Parr, T. and Friston, K. J. (2018). The discrete and continuous brain: from decisions to movement—and back again. *Neural computation*, 30(9):2319–2347.

[113] Pearl, J. (1985). Bayesian netwcrks: A model cf self-activated memory for evidential reasoning. In *Proceedings of the 7th conference of the Cognitive Science Society, University of California, Irvine, CA, USA*, pages 15–17.

[114] Pezzulo, G., Donnarumma, F., Iodice, P., Maisto, D., and Stoianov, I. (2017). Model-based approaches to active perception and control. *Entropy*, 19(6):266.

[115] Pomerleau, D. and Jochem, T. (1996). Rapidly adapting machine vision for automated vehicle steering. *IEEE expert*, 11(2):19–27.

[116] Priorelli, M. and Stoianov, I. P. (2022). Flexible intentions in the posterior parietal cortex: An active inference theory. *bioRxiv*.

[117] Qi, L. (2008). Research on intelligent transportation system technologies and applications. In *2008 Workshop on Power Electronics and Intelligent Transportation System*, pages 529–531. IEEE.

[118] Raza, S., Haider, S., and Williams, M.-A. (2012). Teaching coordinated strategies to soccer robots via imitation. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1434–1439. IEEE.

[119] Regazzoni, C. S., Marcenaro, L., Campo, D., and Rinner, B. (2020). Multisensorial generative and descriptive self-awareness models for autonomous systems. *Proceedings of the IEEE*, 108(7):987–1010.

[120] Rinner, B., Esterle, L., Simonjan, J., Nebehay, G., Pflugfelder, R., Dominguez, G. F., and Lewis, P. R. (2015). Self-aware and self-expressive camera networks. *Computer*, 48(7):21–28.

[121] Rizzolatti, G., Craighero, L., et al. (2004). The mirror-neuron system.

[122] Ross, S. and Bagnell, D. (2010). Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings.

[123] Rouff, C. and Hinchey, M. (2012). *Experience from the DARPA urban challenge*. Springer.

[124] Ruesch, J., Ferreira, R., and Bernardino, A. (2011). A measure of good motor actions for active visual perception. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–6. IEEE.

[125] Sajid, N., Ball, P. J., Parr, T., and Friston, K. J. (2021). Active inference: demystified and compared. *Neural computation*, 33(3):674–712.

[126] Särkkä, S. (2013). *Bayesian filtering and smoothing*. Number 3. Cambridge university press.

[127] Sauer, A., Savinov, N., and Geiger, A. (2018). Conditional affordance learning for driving in urban environments. In *Conference on Robot Learning*, pages 237–252. PMLR.

[128] Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242.

[129] Schneider, F. E. and Wildermuth, D. (2011). Results of the european land robot trial and their usability for benchmarking outdoor robot systems. In *Conference towards autonomous robotic systems*, pages 408–409. Springer.

[130] Schroecker, Y., Vecerik, M., and Scholz, J. (2019). Generative predecessor models for sample-efficient imitation learning. *arXiv preprint arXiv:1904.01139*.

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

[132] Seth, A. K. (2014). The cybernetic bayesian brain. In *Open mind*. Open MIND. Frankfurt am Main: MIND Group.

[133] Singh, S. and Saini, B. S. (2021). Autonomous cars: Recent developments, challenges, and possible solutions. In *IOP Conference Series: Materials Science and Engineering*, volume 1022, page 012028. IOP Publishing.

[134] Smith, R., Kirlic, N., Stewart, J. L., Touthang, J., Kuplicki, R., Khalsa, S. S., Feinstein, J., Paulus, M. P., and Aupperle, R. L. (2021). Greater decision uncertainty characterizes a transdiagnostic patient sample during approach-avoidance conflict: a computational modelling approach. *Journal of Psychiatry and Neuroscience*, 46(1):E74–E87.

[135] Smith, R., Kuplicki, R., Teed, A., Upshaw, V., and Khalsa, S. S. (2020). Confirmatory evidence that healthy individuals can adaptively adjust prior expectations and interoceptive precision estimates. In *International Workshop on Active Inference*, pages 156–164. Springer.

[136] Sucar, L. E. (2015). Probabilistic graphical models. *Advances in Computer Vision and Pattern Recognition. London: Springer London. doi*, 10:978–1.

[137] Sussman, J. S. (2008). *Perspectives on intelligent transportation systems (ITS)*. Springer Science & Business Media.

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

[139] Thornton, C. (2010). Gauging the value of good data: Informational embodiment quantification. *Adaptive Behavior*, 18(5):389–399.

[140] Tschantz, A., Seth, A. K., and Buckley, C. L. (2020). Learning action-oriented models through active inference. *PLoS computational biology*, 16(4):e1007805.

[141] Vogt, D., Ben Amor, H., Berger, E., and Jung, B. (2014). Learning two-person interaction models for responsive synthetic humanoids. *Journal of Virtual Reality and Broadcastings*, 11(1).

[142] Wang, Y., Liu, Z., Zuo, Z., Li, Z., Wang, L., and Luo, X. (2019). Trajectory planning and safety assessment of autonomous vehicles based on motion prediction and model predictive control. *IEEE Transactions on Vehicular Technology*, 68(9):8546–8556.

[143] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3):279–292.

[144] Winfield, A. F. (2014). Robots with internal models: a route to self-aware and hence safer robots.

[145] Xie, G., Li, Y., Han, Y., Xie, Y., Zeng, G., and Li, R. (2020). Recent advances and future trends for automotive functional safety design methodologies. *IEEE Transactions on Industrial Informatics*, 16(9):5629–5642.

[146] Xin, J., Wang, C., Zhang, Z., and Zheng, N. (2014). China future challenge: Beyond the intelligent vehicle. *IEEE Intell. Transp. Syst. Soc. Newslett*, 16(2):8–10.

[147] Yang, Q., Gu, Y., and Wu, D. (2019). Survey of incremental learning. In *2019 chinese control and decision conference (ccdc)*, pages 399–404. IEEE.

[148] Zhang, S., Zhi, Y., He, R., and Li, J. (2020). Research on traffic vehicle behavior prediction method based on game theory and hmm. *IEEE Access*, 8:30210–30222.

[149] Zheng, Y., Zhang, Y., Ran, B., Xu, Y., and Qu, X. (2020). Cooperative control strategies to stabilise the freeway mixed traffic stability and improve traffic throughput in an intelligent roadside system environment. *IET Intelligent Transport Systems*, 14(9):1108–1115.