IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 7, 2022, accepted April 24, 2022, date of publication May 4, 2022, date of current version May 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3172712

Active Inference Integrated With Imitation Learning for Autonomous Driving

SHEIDA NOZARI[1,2,3], (Member, IEEE), ALI KRAYANI[1,3], (Member, IEEE), PABLO MARIN-PLAZA[2], LUCIO MARCENARO[1,3], (Senior Member, IEEE), DAVID MARTÍN GÓMEZ[2], (Member, IEEE), AND CARLO REGAZZONI[1,3], (Senior Member, IEEE)

[1]Department of Engineering and Naval Architecture (DITEN), University of Genoa, 16121 Genoa, Italy
[2]Department of Systems Engineering and Automation, Carlos III University of Madrid, 28903 Madrid, Spain
[3]Italian National Inter-University Consortium for Telecommunications (CNIT), 43124 Parma, Italy

Corresponding author: Sheida Nozari (sheida.nozari@edu.unige.it)

This work was supported in part by the Spanish Government under Grant PID2019-104793RB-C31 and Grant RTI2018-096036-B-C21, and in part by the Comunidad de Madrid under Grant SEGVAUTO-4.0-CM P2018/EMT-4362.

ABSTRACT Classical imitation learning methods suffer substantially from the learning hierarchical policies when the imitative agent faces an unobserved state by the expert agent. To address these drawbacks, we propose an online active learning through active inference approach that encodes the expert's demonstrations based on observation-action to improve the learner's future motion prediction. For this purpose, we provide a switching Dynamic Bayesian Network based on the dynamic interaction between the expert agent and another object in its surrounding as a reference model, which we exploit to initialize an incremental probabilistic learning model. This learning model grows and matures based on the free-energy formulation and message passing of active inference dynamically at discrete and continuous levels in an online active learning phase. In this scheme, generalized states of the learning world are represented as distance-vector, where it is the learner's observation concerning its interaction with a moving object. Considering the distance vector entail intentions, it enables action prediction evaluation in a prospective sense. We illustrate these points using simulations of driving intelligent agents. The learning agent is trained by using long-term predictions from the generative learning model to reproduce the expert's motion while learning how to select a suitable action through new experiences. Our results affirm that a Dynamic Bayesian optimal approach provides a principled framework and outperforms conventional reinforcement learning methods. Furthermore, it endorses the general formulation of action prediction as active inference.

INDEX TERMS Imitation learning, active inference, dynamic Bayesian network, autonomous driving.

I. INTRODUCTION

In recent years, the demand for an intelligent agent (IA) learning by mimicking expert behavior has grown substantially. Advancement in active learning and communication technology has improved learning potentials in IA to make intelligent decisions and adapt and refine actions in various situations. Many future directions in technology rely on the ability of IA to behave as a human would when presented with the same situation. Examples of such fields are intelligent transportation [1], autonomous systems [2]–[4] and sports tracking data [5].

In these applications and many intelligent tasks, we face executing an action given the agent's current state and its surroundings. The number of possible scenarios in a dynamic environment is too large to cover by explicit programming. Also, an efficient IA must be able to handle unobserved scenarios. While such a task may be expressed as an optimization problem, transferring knowledge from an expert agent is more effective and efficient than exploration and learning from scratch [6], [7].

In addition, learning by trial and error requires a supervision signal that indicates the goal of the expected behavior.

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco J. Garcia-Penalvo.

Typically this supervision can come from a reward function which is calculated precisely for each task. So, as the number of actions grows exponentially in a dynamic environment, defining rewards for such problems is difficult, even unknown in some cases. One of the intuitive ways is to mimic an expert behavior by transferring knowledge through observations and by following the demonstrations step-by-step [8]. This paper presents a learning model to perform complex sequences of actions through imitation.

Imitation learning (IL) [9] works by extracting information from the expert agent's behavior and its interaction with the surrounding environment to make a mapping between the observation and demonstrated behavior. Similar to traditional supervised learning, in IL, the instances present pairs of states and actions. If one exists, the state represents the agent's pose and its status with an attractor. Therefore, Markov decision processes (MDPs) [10] are commonly used to represent expert demonstrations in an IL context. The Markov property dictates that the next state only depends on the previous state and action, eliminating the necessity of the earlier states in the state representation [11]. A typical IL procedure encodes the collected expert demonstrations to state-action pairs to use them in policy training. However, a direct mapping between state and action is not enough. It can happen due to some issues such as insufficient demonstrations and performing a different task due to environmental changes, such as obstacles.

Therefore, IL frequently involves another step that requires the learner 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 reinforcement learning (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 [12], which conceptually makes RL superior to IL.

The critical drawbacks of IL are that the policy never exceeds the suboptimal expert performance and the performance of IL is still highly reliant on the quality of the expert policy. As real-world applications often need high sample efficiency, it is crucial to find a way to integrate IL and RL effectively. Several recent efforts have attempted to combine RL and IL [13]–[15]. These approaches incorporate the cost information of the RL problem into the imitation progression, so the learned policy can both improve faster than their RL complements and outperform the expert policy. Despite reports of improved empirical performance, the theoretical understanding of these combined algorithms is still reasonably limited [16], [17].

The integration of both modalities, RL and IL, enables the learning of complex skills from raw sensory observations [18]. RL is widely used for IA learning to develop decision-making sequences that maximize the reward for a future goal. It specifies which states and actions are desirable through sequential interaction with their environment and other agents [19]. While such reward specifications can be sufficient enough to produce optimal behavior, it represents a significant barrier to the broader applicability of RL in complex observations where we have to consider multiple factors that affect the reward signal [20]. Inverse reinforcement learning (IRL) [21] bypasses this issue by assuming that an agent receives the sequences of observation-action tuples. It tries to learn how to map observations to actions from these sequences through estimating a reward function. By approximating this function rather than directly learning the state-action, the apprentice can learn a reward function in new scenarios that explains the observed expert behavior. Moreover, it allows adapting to perturbation in a dynamic environment [22]. However, imitating each step often becomes impracticable when the learning agent and the environment are different from those in the demonstration. Also, using IL to track an agent in motion is still a challenging task. In many cases, the agent does not have to follow the expert unconditionally. Instead, it must care about the demonstrator's intention or the goal-based imitation [23].

In parallel, Active Inference (AIn) [24] suggests a framework where the agent learns to minimize the divergence between expectation and evidence (i.e., surprise or abnormality) by selecting an action based on probabilistic decision making. Surprise is an information-theoretic quantity that can be approximated with variational Free Energy (FE) [25]. FE explains perception, action, and model learning in a Bayesian probabilistic way that provides an upper bound on the negative log-evidence or surprise [26]. The notion of AIn translates predictive coding into an embodied context and argues that surprise (or abnormality) can be minimized in two ways: either by optimizing internal predictions about the world (perception) or via acting on the world to change sensory samples so that they match internal predictions (action) [27].

This work proposes a framework integrating AIn with IL (AIL) for autonomous driving. We provide a hierarchical generative model casting autonomous navigation as minimizing the FE measurements. We illustrate how prediction, perception and navigation emerge from optimizing the hierarchical generative model under active inference. IL is used as a pre-training step to encode an expert demonstration in a coupled Dynamic Bayesian Network (DBN) for a specific task (e.g., overtaking a dynamic obstacle). The coupled DBN is a probabilistic graphical model explaining the dynamic interactions among multiple environmental agents (an expert agent and a dynamic object). Due to its hierarchical nature, DBN can express temporal relationships among high-level variables capturing abstract semantic information about the environment and low-level distributions capturing rough sensory information with their respective evolution through time. Since IL suffers from a fundamental problem known as distributional shift (i.e., distributions over states observed during training are different from those observed during testing), the agent might fail to reach the goal in an

unseen environment. The proposed AIL framework enables the agent to exploit IL by mimicking the expert behaviour under normal circumstances (when predictions match observations), i.e., selecting the same set of actions as the expert and exploring new actions through AIn under abnormal situations. In this work, the exploration phase is intended to select new actions that allow avoiding surprising states in the future, i.e., moving towards the expert reference model. In both exploitation and exploration, the agent aims to occupy environmental states that minimize the FE (i.e., equivalent to maximizing rewards in RL). Consequently, our approach pairs AIn with state-of-the-art machine learning techniques to train an agent that can accomplish its task in a dynamic environment while simultaneously building a generative model of themselves and their surroundings. In other words, AIL maps observed variables and posterior beliefs over latent states and optimal action. The main contributions of this work can be summarized as fourfold:

- Expert demonstrations are explained by a set of configurations encoding the dynamic interaction between moving objects in the environment that facilitates the inference of sensory signals and decision making.
- The exploration-exploitation dilemma is guided by predictive and diagnostic messages. During exploration, the imitator agent learns a new set of configurations and actions incrementally that allow it to come near the expert's reference model.
- The reliance on an explicit reward signal from the environment is unnecessary. The reward is substituted by the FE measurements based on agents' beliefs about environmental states and its actual observations.
- The proposed approach is validated on a real dataset consisting of sensory information collected from two autonomous vehicles. Results show that the proposed approach outperforms conventional RL methods in the number of selected actions, successful travel rate, collision probability, out of boundary probability, and imitation loss.

The remainder of this manuscript is organized as follows. Related work is reviewed in section II. The proposed framework (AIL) is presented is section III. Experimental results are presented in section IV. Finally, conclusions and future directions are drawn in section V.

II. RELATED WORKS

There are different methods for learning a policy from expert demonstrations. Direct learning that learns a supervised model from the demonstrations is the most straightforward way, in which the goal is to learn a mapping from states to actions that mimic the demonstrator [33], [34]. Supervised learning methods are categorized into classification methods when the learner's actions can be classified into discrete classes [35], [36] and regression methods which are used to learn actions in a continuous space [37]. Direct imitation often is not adequate to reproduce suitable behavior due to errors in demonstration [36].

TABLE 1. Abbreviation and mathematical Notations.

Symbol	Description
E	Expert agent
O	Dynamic Object in Expert world
$\hat{\mathrm{O}}$	Dynamic Object in Learner world
L	Learner agent
$\tilde{X}$	Generalized state (GS)
Z	Observations/ input data sequence
H	Observation matrix
k	Time indexes
v	zero-mean Gaussian
A	dynamic matrix
w	noise measurement
$\dot{v}$	Generalized error (GE)
S	Switching vocabulary encoded as super states
s	Cluster of GSs
$\tilde{\mu}$	Generalized mean value
Pos	Position of agent
V	Velocity of agent
$\tilde{\Sigma}$	covariance matrix
D	Identified configuration
B	control model matrix
$\dot{x}, \dot{y}$	velocity components
x, y	position components
d	distance vector
a	Motion of agent
Ω	Anomaly indicator
θ	Angle between the expected and predicted distance vector
$\mathring{\mathrm{D}}$	Activated reference configuration
$\in$	Exploration rate
α	Maximum weight of particles at each instant
β	Associated configuration to α
W	Particle's weight
λ	Message passing
π	Predictive message
$\dot{F}$	Free Energy measurement at the continuous level
$\ddot{F}$	Free Energy measurement at the discrete level
G	Global Free Energy
Q^*	Estimation of Learner's motion
η	Learning rate
γ	Discount factor

Besides, indirect learning can complement direct learning by refining the policies based on expert demonstrations and learner experiences to be more accurate in unseen scenarios. The crucial role of RL to minimize the distinction in IL, which is known as probabilistic inference, have been discussed extensively in literature [38]–[41]. Most of the existing methods developed RL approaches based on different divergence metrics to show optimal control that can be formulated as probabilistic inference in a graphical model to minimize divergence between reward and policy distributions over trajectories [39], [42]. Motion prediction plays a prominent role in maximizing the convergence between expectation and evidence. An efficient prediction provides the agent with the ability to learn appropriate transitions to reach the target autonomously [43]. In [44], a motion planning approach is proposed to avoid collisions by using a specified sparse reward function. It leads to inefficient learning through facing dynamic obstacles if the learning agent does not observe enough to reinforce its actions based on the changes in the environment due to the obstacle's motion. Our work draws significant inspirations from these prior works in RL and aims

TABLE 2. Comparison with existing methods from literature.

functionalities	AIL	[28]	[29]	[30]	[31]	[32]	[33]	[34]	[35]
indirect learning	✓	✓	✓	✗	✗	✗	✗	✗	✗
self-refinement	✓	✓	✓	✓	✓	✓	✗	✗	✗
employing discrete level of beliefs	✓	✓	✓	✓	✓	✓	✓	✓	✓
employing continuous level of beliefs	✓	✗	✗	✗	✗	✗	✓	✓	✗
incremental learning	✓	✓	✓	✗	✓	✗	✗	✓	✗
calculate the future expected reward/FE	✓	✗	✗	✗	✗	✗	✗	✗	✗

TABLE 3. Comparison with existing methods regarding the way of learning and planning. The proposed method learns to plan an agent's motion in a dynamic environment by exploiting an expert agent and exploring new experiences.

method	expert demonstration	trial and error	observation
[37] supervised learning	✓	✗	✓
[42] reinforcement learning	✗	✓	✗
AIL	✓	✓	✓

to provide a probabilistic dynamic learning model which can anticipate future changes in the environment. IRL algorithms have shown promising results in defining a policy to minimize the cost functions or maximizing the entropy of the distribution on state-actions under the learned policy [45]. Early works in IL through IRL operated by matching desired features between policies and expected expert demonstration [29], [46]. Furthermore, Energy-Based IL [30], [40] is an IRL framework that estimates unnormalized probability energy of expert's occupancy measure through score matching, then takes the energy to construct a reward function as a guide for learning the desired policy. Recent expandable approaches to Max-Ent IRL [28], [31], [32], motivated by adversarial approaches to generative modelling [47], present a common view of IL through divergence minimization perspective.

Our work generalizes the objectives based on insight from minimizing free energy [48] and further provides a learning model based on probabilistic interaction between the agents in a dynamic environment. Crucially, the proposed framework enables the imitator agent to infer latent state while learning to infer hierarchically concerning the FE functional. Table 2 and Table 3 summarize the comparison between the proposed framework (AIL) and some existing methods from the literature.

III. PROPOSED FRAMEWORK

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 encoding the dynamic interaction between an *Expert agent* (E) and a *dynamic Object* (O). Consequently, we provide the *Learner agent* (L) a First-Person (FP) model, 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 (AFP) model that L can use to update its knowledge while interacting with an object ($\hat{O}$) in a continuous dynamic environment. All of the mentioned models (i.e., situation, FP and AFP) are Probabilistic Graphical Models (PGMs) that employ graph-based representation to encode various multi-dimensional random variables and represent causal relationships among them [49].

In this work, we propose to use a particular type of PGMs, namely, the Dynamic Bayesian Network (DBN) [50]. Due to its hierarchical nature, DBN 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 [51]. Since the network size increases over time, performing inference using the entire network would be intractable for all but trivial time duration. Fortunately, efficient recursive algorithms have been developed to perform exact inference on specific types of DBNs, or approximate inference on more general DBNs' varieties [52], [53]. Recent works studied several algorithms for inference in PGMs following a data-driven approach [54], [55]. A modern inference mechanism, namely, the Markov Jump Particle Filter (MJPF) presented in [54] can be employed to facilitate the generation of behavior based on DBN models learned computationally from data.

A. OFFLINE LEARNING PHASE

The aim of the offline learning process is to learn the situation model (i.e., the reference model that L can use for initialization) based on the E behavior. Initialization is conducted by mapping the reference DBN structure onto the L moving reference system as an FP reference model.

1) SITUATION MODEL

The situation model consists of a switching DBN [56] representing the interaction of two dynamic entities, E and O. 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 agents' (E and O) observations are represented by variables Z_k^E and Z_k^O, respectively (① in Fig. 1). At a higher level, hidden continuous Generalized States (GSs) [57] can be formed describing the agents' instantaneous dynamics up to a chosen

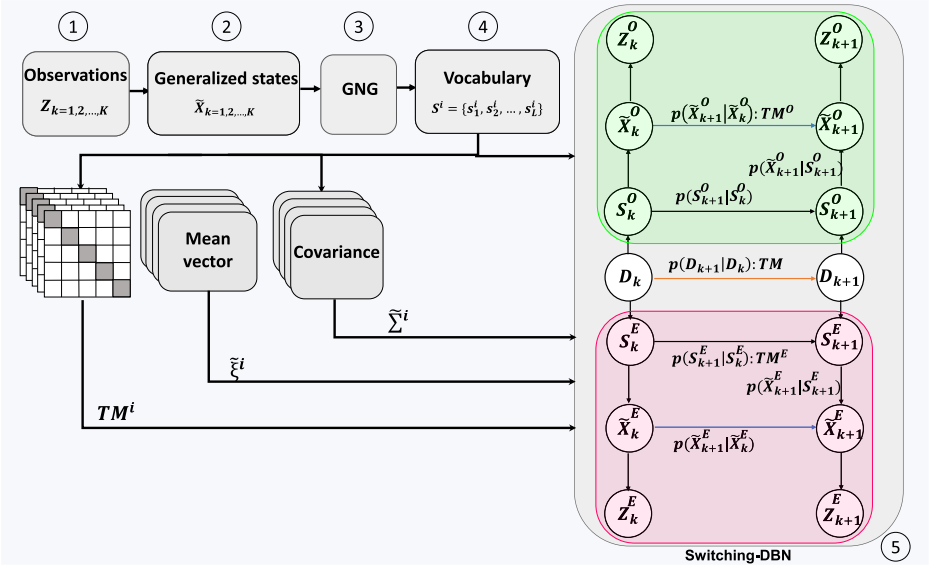


FIGURE 1. An overview of learning a Dynamic Interaction Model. The arrows in switching DBN represents the conditional probabilities between involved variables. Vertical arrows facilitates the causalities description between continuous and discrete levels of inference and observed measurements. Horizontal arrows explain temporal causalities between hidden variables. In particular, the orange arrow encodes the interaction of couples of agents, and blue arrows represent the influence at a continuous level.

n-th order temporal derivative. Thus, a joint GS ($\tilde{\mathrm{X}}_k$) (② in Fig. 1) incorporating the dynamics of multiple agents (i.e., E and O) at each time instant k can be defined as follows:

$$\tilde{\mathrm{X}}_k = [\tilde{\mathrm{X}}_k^{\mathrm{E}} \ \tilde{\mathrm{X}}_k^{\mathrm{O}}]^{\mathsf{T}}, \tag{1}$$

where $\tilde{\mathrm{X}}_k^{\mathrm{E}}$ and $\tilde{\mathrm{X}}_k^{\mathrm{O}}$ denote the GSs of E and O, 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 $\tilde{\mathrm{X}}_k^i = [\boldsymbol{x} \ \dot{\boldsymbol{x}}]^{\mathsf{T}}$ where $\boldsymbol{x} \in \mathbb{R}^d, \dot{\boldsymbol{x}} \in \mathbb{R}^d, i \in \{\mathrm{E}, \mathrm{O}\}$ and d stands for the dimensionality of the state vector. Each observed sensor variable Z_k^i is assumed to be related with the corresponding agent's hidden state variable $\tilde{\mathrm{X}}_k^i$ by a linear relationship according to the following observation model:

$$\mathrm{Z}_k^i = \mathrm{H}\tilde{\mathrm{X}}_k^i + v_k, \tag{2}$$

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:

$$\tilde{\mathrm{X}}_k^i = \mathrm{A}\tilde{\mathrm{X}}_{k-1}^i + w_k, \tag{3}$$

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, namely the Null Force Filter (NFF). An NFF can be interpreted as a generalized Kalman Filter (KF) [58], which uses the innovations obtained by observing an input data sequence Z_k^i to estimate a new situation model 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:

$$\dot{\boldsymbol{v}} = \mathrm{H}^{-1}\big(\mathrm{Z}_k^i - \mathrm{H}\tilde{\mathrm{X}}_k^i\big). \tag{4}$$

The couples $(\tilde{\mathrm{X}}^i, \dot{v})$ obtained by NFF along the interaction time series are defined as generalized errors (GEs).
Those GEs can be clustered using an unsupervised method. We employ the Growing Neural Gas with utility measurement (GNG-U) [59] which outputs a set $\mathbf{S}^i$ of (switching) discrete variables (i.e., clusters) representing the discrete level of the switching DBN (⑤ in Fig. 1). Each cluster describes in which region of the GS space, with which difference in the dynamic motion (with respect to 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 O, 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:

$$\mathbf{S}^i = \{\mathrm{s}_1^i, \mathrm{s}_2^i, \dots, \mathrm{s}_{\mathcal{L}_i}^i\}, \tag{5}$$

where $\mathcal{L}_i$ is the total number of clusters associated with agent i and $\mathrm{s}_l^i \in \mathbf{S}^i$ 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 value $\tilde{\boldsymbol{\mu}}^{\mathrm{s}^i} = [\mu_{Pos}^{s^i}\ \mu_V^{s^i}]$, 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 $s_k^i \in \boldsymbol{S}^i$. Joint active superstates from different agents occurring simultaneously form an interaction configuration defined as $\mathrm{D}_k = [\mathrm{s}_k^{\mathrm{E}}, \mathrm{s}_k^{\mathrm{O}}]^{\intercal}$. Consequently, an additional vocabulary of dictionary configurations can be defined and included in the DBN at a higher hierarchical level, such that:

$$\mathbf{D} = \{\mathrm{D}_1, \mathrm{D}_2, \cdots, \mathrm{D}_{\mathrm{M}}\}, \tag{6}$$

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

$$\mathrm{D}_m = \left[(\mu_{Pos}, \mu_V)^{\mathrm{E}}, (\mu_{Pos}, \mu_V)^{\mathrm{O}}\right]. \tag{7}$$

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 (2). 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:

$$\tilde{\mathrm{X}}_k^i = \mathrm{A}\tilde{\mathrm{X}}_{k-1}^i + \mathrm{B}\mu_V^{s_k^i} + w_k, \tag{8}$$

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

$$\mu_V^{s_k^i} = [\dot{x}_{\mathrm{s}_k^i}, \dot{y}_{\mathrm{s}_k^i}], \tag{9}$$

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 (8) corresponds to a cluster dependent motivated dynamics whose effects are encoded in $\mu_V^{s_k^i}$ 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 (TM). Learning the TM 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:

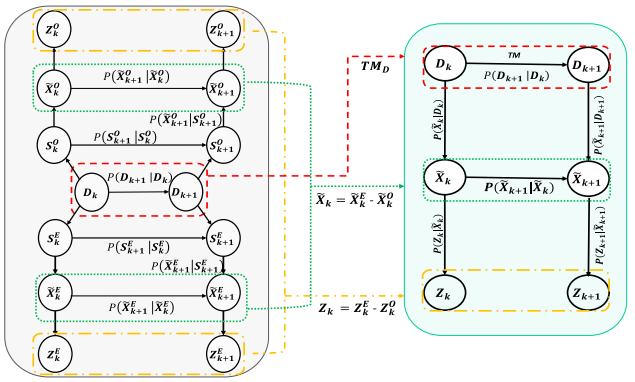


FIGURE 2. Initializing the first person model (right side) by exploiting the situation model (left side). The situation model shows the switching DBN from Fig. 1. The first-person model arrows represent conditional probabilities: vertical arrows introduce causalities between both (discrete and continuous) levels of influence and observed measurements. Horizontal arrows explain temporal causalities between hidden variables.

$$\mathrm{TM} = \begin{bmatrix} \mathrm{P}(\mathrm{D}_1|\mathrm{D}_1), & \mathrm{P}(\mathrm{D}_1|\mathrm{D}_2), & \ldots, & \mathrm{P}(\mathrm{D}_1|\mathrm{D}_{\mathrm{M}}) \\ \mathrm{P}(\mathrm{D}_2|\mathrm{D}_1), & \mathrm{P}(\mathrm{D}_2|\mathrm{D}_2), & \ldots, & \mathrm{P}(\mathrm{D}_2|\mathrm{D}_{\mathrm{M}}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{P}(\mathrm{D}_{\mathrm{M}}|\mathrm{D}_1), & \mathrm{P}(\mathrm{D}_{\mathrm{M}}|\mathrm{D}_2), & \ldots, & \mathrm{P}(\mathrm{D}_{\mathrm{M}}|\mathrm{D}_{\mathrm{M}}) \end{bmatrix}, \tag{10}$$

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

2) FIRST PERSON MODEL

The First Person (FP) model can be seen as a situation model transformed in such a way that allows a learning agent L to directly use its own observations and generate state series describing its relative state with respect to another interacting dynamic agent $\hat{\mathrm{O}}$. It provides L agent with the capability to imitate the expert motions by generating transformed sequences from the situation model (Fig. 2). A mapping implies defining all DBN nodes of the new FP model (discrete and continuous) and probabilistic dependency models starting from the situation model nodes and links. Therefore, The FP model can be considered as an initialization generative switching model represented by a Generalized DBN (GDBN), which can be used to predict interaction states under the perspective of a learning agent.

a: FP MODEL INITIALIZATION

The FP model depicted in Fig. 2-(right side) is initialized to allow L agent to exploit the switching DBN corresponding to the situation model (i.e., pure IL from expert demonstrations). The discrete variables at the configuration level (top level of the FP model) represent the learned set of configurations $\mathrm{D}_m \in \mathbf{D}$. In the FP model, 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 TM model can characterize the temporal dependencies of discrete series

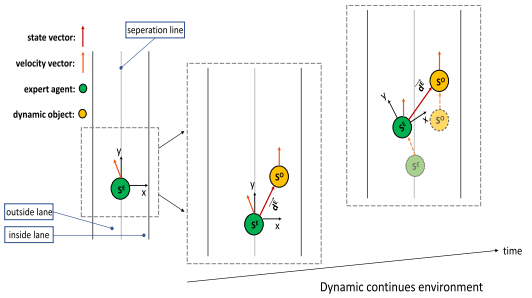


FIGURE 3. Calculating the expert distance vector (d_z^E) during the agents movements in a non stationary continues environment.

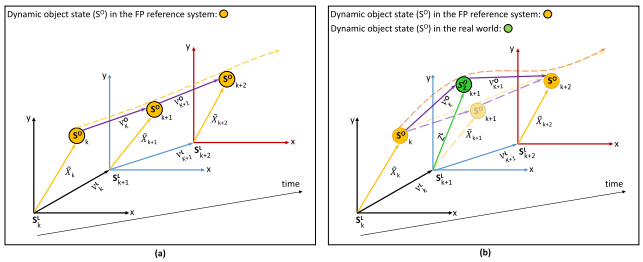


FIGURE 4. The figure shows the learner agent movements in a continuous dynamic environment by the estimated motion at each time. The learner state (s^L) at each time instant (k) is the origin of the measurements that the velocity vector (v^L) leads it to the next state (s^L_{k+1}). a) It shows a normal situation that the learner's interaction with the another dynamic object (s^O) is similar with the FP model's prediction. b) It shows an abnormal situation, where the prediction ($\tilde{X}$) 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.

of interactions in the FP model among L and $\hat{O}$ for the experiences to be imitated. Thus, the transition model can be directly mapped onto the FP model from the corresponding TM of the situation model (i.e., $TM_D = TM$).

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

$$\tilde{X}_k = \left[\tilde{X}_k^E - \tilde{X}_k^O\right] = \left[(\boldsymbol{x}^O - \boldsymbol{x}^E), (\dot{\boldsymbol{x}}^O - \dot{\boldsymbol{x}}^E)\right]. \quad (11)$$

The relative positions of E and O in the situation model are illustrated in Fig. 3. The relative distance vector is highlighted as the difference in absolute coordinates and velocities of the two objects. The distance vector in the FP model is shown in Fig. 4 where the relative learner reference system is depicted to highlight the information captured in the FP model. Moreover, the observation (Z_k) of L and $\hat{O}$ can be mapped onto observations (Z_k^i) of both agents,(E, O) according to the following equation:

$$Z_k = \left[Z_k^E - Z_k^O\right]. \quad (12)$$

To this end, a configuration $D_m \in \mathbf{D}$ at the discrete level of the FP model is represented by a joint superstate of each agent at time instant k, i.e., $D_k = \left[s_k^L, s_k^O\right]$. Thus, the model can predict the expected future configurations based on the dynamic transition rules encoded in the transition matrix TM_D and predict GSs based on the following dynamic model:

$$\tilde{X}_k = A\tilde{X}_{k-1} + B\mu_V^{D_k} + w_k, \quad (13)$$

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

B. ONLINE ACTIVE LEARNING PHASE

In this section, we propose a hybrid mechanism allowing L agent to learn how it should behave in a dynamic environment by integrating imitation learning with active inference [60]. The FP model can be used during the active learning stage providing suitable predictions and learning policies to learn the best set of actions that L agent should take. However, the FP model must be integrated with active states describing how the agent can act in the environment to change sensory signals in order to match internal predictions of the FP model and imitate efficiently.

1) ACTIVE FIRST PERSON MODEL

During active learning, L agent has to interact with another dynamic agent $\hat{O}$ in real-time. L starts assessing the current situation and evaluates if E has experienced the same situation by relying on the FP model that encodes the dynamic interaction between E and O. If L is facing the same situation faced by E, then L tries to imitate the same actions E has performed, which are captured in the FP model (i.e., imitation learning). L can observe $\hat{O}$ by its exteroceptive sensor that provides the relative distance Z_k (a vector incorporating the differences in positions and velocities) between the current origin of the L reference system and the other agent $\hat{O}$ in the environment. The advantage of using the relative distance is that it provides a generalizable solution that can be employed in different environments.

The active agent (L), maintains an internal generative model $P(Z, \tilde{X}, D, a)$ of the prevalent environment expressed in an Active-FP (AFP) model (as depicted in Fig. 5) and aims to minimize implicitly the difference between what it believes about the environmental states and what it perceives. The AFP model specifies the joint probability of observations ($\mathbf{Z}$), their hidden causes ($\tilde{\mathbf{X}}$, $\mathbf{D}$) and actions ($\boldsymbol{a}$). Since the environment is modelled as a Markov Decision Process (MDP), the AFP model can be factorized as:

$$P(\mathbf{Z}, \tilde{\mathbf{X}}, \mathbf{D}, \boldsymbol{a}) = P(D_1)P(\tilde{X}_1)\left[\prod_{k=2}^{T} P(Z_k|\tilde{X}_k) P(\tilde{X}_k|\tilde{X}_{k-1}, D_k, a_{k-1})P(a_{k-1}|D_{k-1})\right]. \quad (14)$$

The proposed AIn approach integrated with IL (**AIL**) involves three main steps: 1) Prediction and perception, 2) Action selection and 3) FE calculation and Action updates.

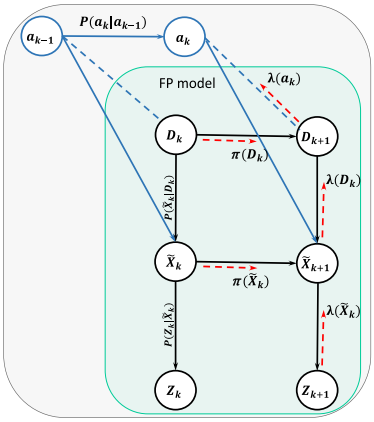


FIGURE 5. Active First Person model. To run an online learning procedure, the model applies the learner agent's motions (a) to the FP model at each time instant (blue arrows).

2) PREDICTION AND PERCEPTION

L employs Particle Filter (PF) to predict the configurations D_k (visited by E) and consequently estimate the relative distance $\tilde{\mathrm{X}}_k$ from $\hat{\mathrm{O}}$ 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 $\hat{\mathrm{O}}$ and the expected configuration (D_1). In the successive iterations ($k > 1$), L relays on the interactive transition matrix TM 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 TM 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:

$$\tilde{\mathrm{X}}^{\mathrm{E}}_{k,n} = \mu^{\mathrm{E}}_{\mathrm{D}_{k,n}} + w_k, \tag{15}$$

$$\tilde{\mathrm{X}}^{\mathrm{O}}_{k,n} = \mu^{\mathrm{O}}_{\mathrm{D}_{k,n}} + w_k, \tag{16}$$

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:

$$\tilde{\mathrm{X}}_{k,n} = \tilde{\mathrm{X}}^{\mathrm{O}}_{k,n} - \tilde{\mathrm{X}}^{\mathrm{E}}_{k,n}. \tag{17}$$

Thus, this approximation depends on the hypothesised configuration that explains implicitly the conditional probability $\mathrm{P}(\tilde{\mathrm{X}}_{k,n}|\tilde{\mathrm{X}}_{k-1,n}, D_{k,n})$. In this sense, L associates itself to a specific configuration ($\mathrm{D}_{k,n}$) and predicts the relative distance from the current dynamic object $\hat{\mathrm{O}}$ 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 L and $\hat{\mathrm{O}}$ - L can evaluate if the situation it is experiencing has already faced by E in order to make decision on actions (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 the AFP 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:

$$\Omega_{k,n} = \cos(\theta) = \frac{\tilde{\mathrm{Z}}_k \ . \ \tilde{\mathrm{X}}_{k,n}}{||\tilde{\mathrm{Z}}_k|| \ ||\tilde{\mathrm{X}}_{k,n}||}. \tag{18}$$

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:

$$\lambda(\mathrm{D}_k) = \lambda(\tilde{\mathrm{X}}_k)\mathrm{P}(\tilde{\mathrm{X}}_k|\mathrm{D}_k), \tag{19}$$

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:

$$W_{k,n} = W_{k,n} \times \lambda(\mathrm{D}_k). \tag{20}$$

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 (ρ). The former is defined as:

$$\epsilon_k = 1 - \alpha_k, \tag{21}$$

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:

$$\alpha_k = \max_n W_{k,n}, \tag{22}$$

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 define ρ 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:

$$a_k \sim \begin{cases} \mu_V^{\mathrm{D}_k^\beta} = \arg\max_{a_k} \mathrm{Q}(\mathcal{A}, \mathrm{D}_k^\beta), \ \text{if } \epsilon < \rho \ \text{(exploitation)}, \\ \textit{random} \text{ from } \mathcal{A}^+, \ \text{if } \epsilon \geq \rho \ \text{(exploration)}, \end{cases} \tag{23}$$

where a_k are the active states (i.e., actions) realizing the top level of the AFP model, $\mathcal{A} = \{\mathcal{A}^{\mathrm{E}}, \mathcal{A}^+\}$, where

$\mathcal{A}^{\mathrm{E}} = \{a_1^{\mathrm{E}}, a_2^{\mathrm{E}}, \ldots, a_{\mathrm{Y}}^{\mathrm{E}}\}$ is a set of actions performed by the expert (E) and encoded in the situation model that the learner aims to imitate during exploitation and $\mathcal{A}^{+} = \{a_1, a_2, \ldots, a_8\}$ 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 (22) defined as:

$$\beta = \arg\max_{n} (W_{k,n}). \tag{24}$$

Moreover, during exploration, L saves the new configurations D_k^{+} (not seen by E) it is experiencing along with the performed actions $a_k^{+} \in \mathcal{A}^{+}$ in a set ($\mathcal{C}$). After finishing a certain experience L clusters all the pairs $[\mathrm{D}_k^{+}, a_k^{+}]$ saved in $\mathcal{C}$ 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:

$$\mathrm{Q} = \begin{bmatrix} \mathrm{P}(a_1^{\mathrm{E}}|\mathrm{D}_1) & \ldots & \mathrm{P}(a_1^{\mathrm{E}}|\mathrm{D}_{\mathrm{M}}) & \mathrm{P}(a_1^{\mathrm{E}}|\mathrm{D}^{++}) & \ldots \\ \mathrm{P}(a_2^{\mathrm{E}}|\mathrm{D}_1) & \ldots & \mathrm{P}(a_2^{\mathrm{E}}|\mathrm{D}_{\mathrm{M}}) & \mathrm{P}(a_2^{\mathrm{E}}|\mathrm{D}^{++}) & \ldots \\ \vdots & \ddots & \vdots & \vdots & \\ \mathrm{P}(a_{\mathrm{Y}}^{\mathrm{E}}|\mathrm{D}_1) & \ldots & \mathrm{P}(a_{\mathrm{Y}}^{\mathrm{E}}|\mathrm{D}_{\mathrm{M}}) & \mathrm{P}(a_{\mathrm{Y}}^{\mathrm{E}}|\mathrm{D}^{++}) & \ldots \\ \mathrm{P}(a^{++}|\mathrm{D}_1) & \ldots & \mathrm{P}(a^{++}|\mathrm{D}_{\mathrm{M}}) & \mathrm{P}(a^{++}|\mathrm{D}^{++}) & \ldots \\ \vdots & & \vdots & \vdots & \end{bmatrix}, \tag{25}$$

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 update the transition model defined in (10) by adding new rows and columns which are related to the new configurations incrementally.

In *Exploitation*, if the current configuration is an observed one by E, then L takes the adapted expert action from the FP model by activating the most similar reference configuration ($\mathrm{D}_k = \mathrm{D}_k^{\beta}$) 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.

Besides, in *Exploration*, if the mismatch between predictions and observation is too high, the model can not apply the direct imitation of E agent by taking a learned action from the situation model. However, if L agent 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).

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

Algorithm 1 Active Inference Integrated With IL (AIL)

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

The AFP model improves the L's behavior during the training by minimizing the divergence between the situation model and the AFP model 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, 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 on a stochastic task while switching between exploration and exploitation.

4) IMITATION COST

The IA faces multiple tasks that need ample action space. Therefore it is challenging to acquire an appropriate policy by a onefold cumulative reward strategy. Hence, we cannot just rely on a cumulative reward strategy to train the IA to learn effective behavior policies. This work suggests training the learner agent by imitating expert manipulations to solve this problem effectively by employing IRL to IL, which postulates that expert behavior is to optimize the expected motion of the learner agent over time. Most IRL methods formalize the underlying decision-making problem as an MDP, a model of a discrete-time process wherein an agent's actions may stochastically influence its environment. IRL 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) ensures a dynamic equilibrium between L and its prevalent environment. The FE measurements are based on the AFP hierarchy's messages (messages passing from top-to-down and bottom-to-up). The message (λ) passing from lower nodes to upper nodes (see Fig. 5) have 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 detecting whether new observations are similar to previously learned situations encoded in the FP model. Suppose predictions from the FP model are not compliant with observations, then the 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 the prior policy and posterior policy at both continuous and discrete levels. The goal is to allow L maximizing 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 (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.

a: FE MEASUREMENT AT THE CONTINUOUS LEVEL

The AFP model allows evaluating how much the sensory measurements support predictions and thus evaluating if the selected actions were good or bad by relying on the FE. The FE at the continuous level 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 an action a_{k-1} under both exploration and exploitation. Thus, the taken action (a^L_{k-1}) guides the system to calculate the expected FE [61] at the continuous level ($\dot{F}$) based on the Kullback Leibler-Divergence ($\mathcal{D}_{\mathcal{KL}}$) [62] between $\pi(\tilde{\mathrm{X}}_k)$ and $\lambda(\tilde{\mathrm{X}}_k)$. Hence, the expected FE can be expressed as:

$$\dot{F} = \mathcal{D}_{\mathcal{KL}}\Big(\lambda(\tilde{\mathrm{X}}_k)||\pi(\tilde{\mathrm{X}}_k)\Big) = \int \lambda(\tilde{\mathrm{X}}_k)\log\left(\frac{\lambda(\tilde{\mathrm{X}}_k)}{\pi(\tilde{\mathrm{X}}_k)}\right)\mathrm{d}\tilde{\mathrm{X}}_k. \tag{26}$$

Our goal is to find a policy, such that the learner's behavior matches the reference demonstrations. For this purpose, our objective is to minimise 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 allowing it to approach towards the E's reference vocabulary as soon as possible during exploration.

b: FE MEASUREMENT AT THE DISCRETE LEVEL

The FE at this level ($\ddot{F}$) is computed by employing the Mahalanobis distance ($\mathcal{D}_{\mathcal{M}}$) [63] 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 μ^D_V, defined as:

$$\ddot{F} = \mathcal{D}_{\mathcal{M}}\big(a^L_k, a^{\mathrm{E}}_k\big), \tag{27}$$

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

c: GLOBAL FE

The Global FE ($\mathcal{G}$) is based on the losses computed at both continuous and discrete levels (defined in (26) and (27)). If L agent is in a observed configuration (exploitation case) the GFE is defined as:

$$\mathcal{G} = \dot{F}. \tag{28}$$

Otherwise, if it experiences a new configuration or improving the action selection regarding the recorded explored states, the GFE can be expressed as:

$$\mathcal{G} = \mathbb{E}(\dot{F}, \ddot{F}). \tag{29}$$

5) ACTION UPDATE

The AFP model takes advantage of both discrete and continuous level dynamically to decrease the imitation loss by improving the action selection through the online learning procedure. Our objective is to minimise the long term cost by taking down the global FE measurements defined in (28) and (29). L adapts the action selection process by updating the Q-table defined in (25) based on the global FE and according to:

$$\mathrm{Q}^*_k = (1-\eta)\mathrm{Q}(\mathrm{D}_{k-1}, a_{k-1}) + \eta\left[(1-\mathcal{G}) + \gamma \max_{a_k} \mathrm{Q}(D_k, a_k)\right], \tag{30}$$

function FREE ENERGY MEASUREMENT(Q, a_k^L, D_k)
  $a_k^E \sim max\ Q(:, D_k)$
  $S_k^E \sim D_k \leftarrow$ 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
  $\hat{v}_k \sim (V_k^L - \mu_{V,k}^E)$
  $\ddot{F} = \mathcal{D}_{\mathcal{M}}(a_k^L, a_k^E) = \sqrt{\hat{v}_k^T (\Sigma_k^E)^{-1} \hat{v}_k} \leftarrow$ The FE at current time
  Calculate $\dot{F}$ using (26) $\leftarrow$ The excepted future FE
  if the L is in exploration: **then**
    $\mathcal{G} = \mathbb{E}(\dot{F}, \ddot{F}) \leftarrow$ Global Free Energy
  else
    $\mathcal{G} = \dot{F} \leftarrow$ Global Free Energy
  end if
  Update Q^* using (31)
  return Q^*
end function

where η is the learning rate which controls how quickly the learning agent adopts to the explorations imposed by the environment, $\mathcal{G}$ 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. Since the Q table used in this work is a probabilistic table, (30) can be rewritten in probabilistic form as follows:

$$Q^* = (1-\eta)P(a_{k-1}|D_{k-1}) + \eta\left[(1-\mathcal{G}) + \gamma \max_{a_k} P(a_k|D_k)\right]. \quad (31)$$

IV. EXPERIMENTS AND RESULTS

A. EMPLOYED 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' [64]. The vehicles positional information and the corresponding velocities are obtained from the odometry module. This work considers two scenarios:

- **lane-keeping scenario (following behavior):** iCab 2 follows another agent (iCab 1) as shown in Fig. 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):** iCab 2 overtakes iCab 1 (considered as a dynamic obstacle) to change the lane without collision. This scenario consists of two cases, overtaking from the left side and overtaking from the right side as depicted in Fig. 8-(b) and Fig. 8-(c). In this scenario, iCab 2 has a higher speed than iCab 1.

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 situation model that the learner agent L will use to imitate E.

B. OFFLINE LEARNING PHASE

This section shows the process of learning the situation model from 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) 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 (6)) that encode the dynamic interaction among the two agents. The total number of clusters and configurations is 36 each. Fig. 9-(a)-(b)-(c) illustrates the clusters and configurations learned in different scenarios and Fig. 9-(d)-(e)-(f) shows the corresponding transition matrices.

FIGURE 6. Autonomous vehicles: iCab 1 and iCab 2.

FIGURE 7. The yellow parts shows the experimental zone.

C. ONLINE LEARNING PHASE

1) EXPERIMENTAL SETTING

During the online active learning phase, the AFP model relies on the FP, which has been initialized using the situation model. Thus, the discrete level in the three models represents the learned configurations during the offline phase. The initial Q table contains only the learned configurations and it is defined as follows:

$$Q = \begin{array}{c} \\ a_1^E \\ a_2^E \\ \vdots \\ a_{36}^E \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & \dots & D_{36} \end{array} \\ \begin{bmatrix} \frac{1}{36} & \frac{1}{36} & \cdots & \frac{1}{36} \\ \frac{1}{36} & \frac{1}{36} & \cdots & \frac{1}{36} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{36} & \frac{1}{36} & \cdots & \frac{1}{36} \end{bmatrix} \end{array}$$

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 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 $\hat{O}$ in a dynamic environment. Each episode consists of 10 iterations,

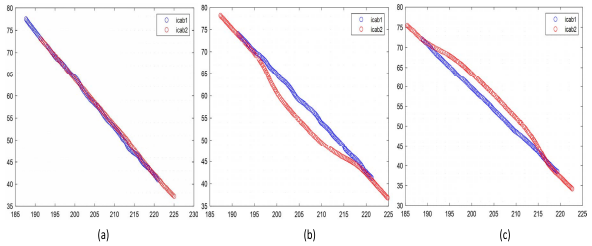


FIGURE 8. iCabs interactions. a) iCab 2 follows iCab 1, b) iCab 2 overtakes iCab 1 from the left side, and c) iCab 2 overtakes iCab 1 from the right side.

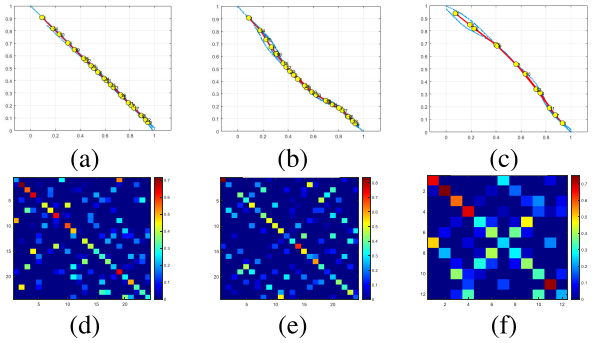


FIGURE 9. Learning the situation model. a) Clustering of GEs in the lane-keeping scenarios, b) Clustering of GEs in the lane-changing from the left scenario. c) Clustering of GEs in the lane-changing from the right scenario. sub-Fig. (d)-(e)-(f) are the corresponding transition matrices to sub-Fig. (a)-(b)-(c), respectively.

i.e., L tries 5k iterations by 500 different start positions to learn the policies.

a: ESTIMATE THE SAFE DISTANCE FOR THE LEARNER AGENT

The FE measurement at the continuous level helps the learner determine a safe distance from the moving object. The safe distance expresses the possibility that the learner agent can continue lane-keeping without collision probability. At each time instant, the system finds the minimum and the mean value of $\dot{F}$, which is calculated by the KL divergence defined in (26). 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 agent, which is changed dynamically at each time instant during the learning phase until the completion of training. The 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 from the learned tables. We evaluate the performance of the proposed method in different experiments and compare it with four learning algorithms from the literature, 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 involves two main issues, action selection and imitation loss.

2) ACTION SELECTION

L predicts the configurations (D_m) visited by E by employing PF and then estimates the relative distance from $\hat{O}$ 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 TM (at the first time instant $k = 1$, PF generates 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 dynamic obstacle). The number of taken actions describes the effort made by the agent to reach the goal. A good policy requires fewer actions and less time to reach the goal, while a lousy policy requires more actions and time.

Fig. 10-(a) shows the mean of taken actions 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 less actions compared to other methods. This can be explained by the fact that initializing the FP model using the situation model can decrease the exploration rate. Moreover, exploiting from sub-optimal expert demonstrations at similar states plays a vital role to driving in a shorter time than exploring the environment from scratch. The threshold ρ has a great impact on the exploration rate, we train 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. 11. In addition, updating particles' weights to adjust the action selection procedure allows L avoiding abnormalities and adapting to new experiences.

Fig. 12 demonstrates how the exploitation and exploration rates affect the FE during the learning phase. Refining the action selection can adapt to new experiences and minimise the FE. Balancing exploration and exploitation is one of the most challenging tasks in RL. The imbalance between exploration and exploitation might lead to adverse effects on learning performance. On the one hand, the domination of exploration would obstruct the agent to maximize short-term reward, i.e., explorative 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. 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. 14 and 15). Improving the action selection skill leads L to perform more successful movements in the dynamic environment as shown in Fig. 16.

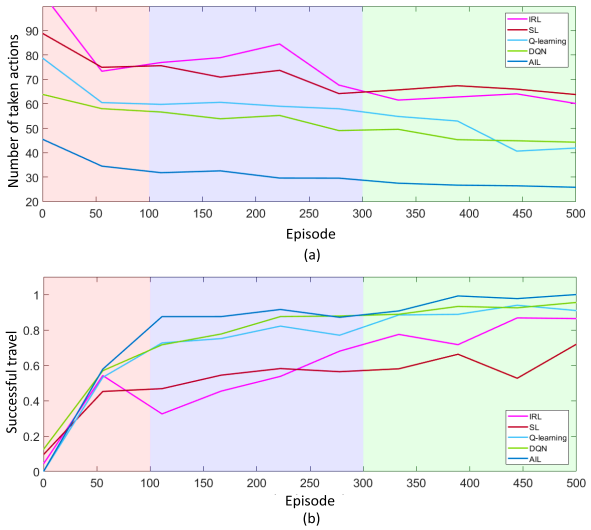


FIGURE 10. The learner performance. a) The number of taken actions during learning phase. b)Success rate.

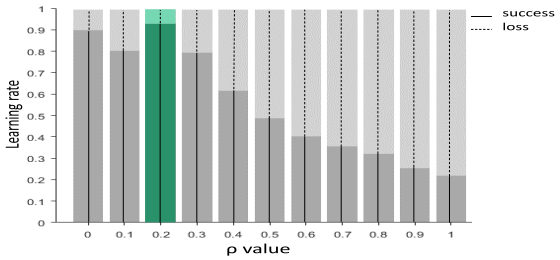


FIGURE 11. ρ is a threshold that plays the control role to separate the exploration and exploitation mode. We trained the model with different ρ values to find the the most suitable one by trail and error. The green bar is the selected one.

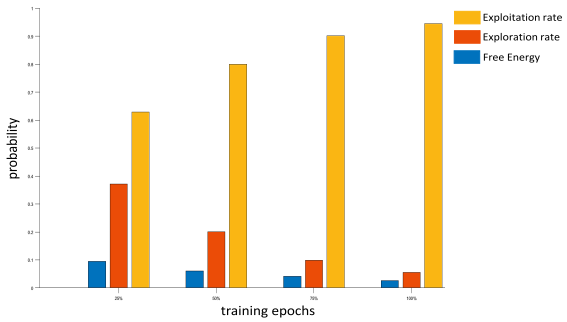


FIGURE 12. Illustration of the exploitation rate and exploration rate after each training quarter and their effect on the FE.

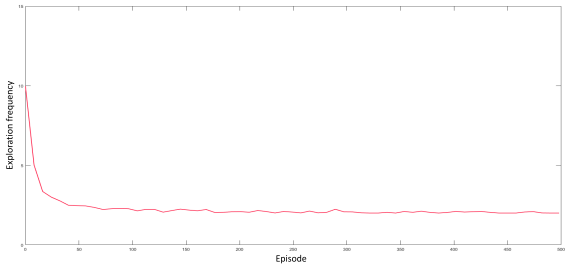


FIGURE 13. Exploration frequency. It shows after how many explored action the learner goes back to the exploitation mode (average number for each episode).

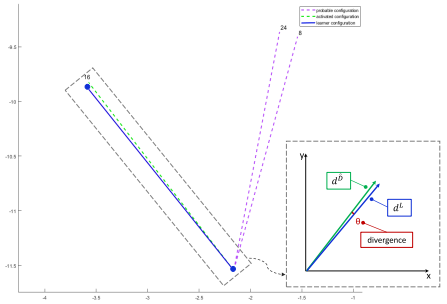


FIGURE 14. An example illustrates when the learner agent is in the exploitation mode. Purple lines show the relative distance from the most probable configuration, while the green line represents the relative distance from the activated configuration. The learner exploits the activated configuration, leading to a lower divergence (θ) between blue and green distances).

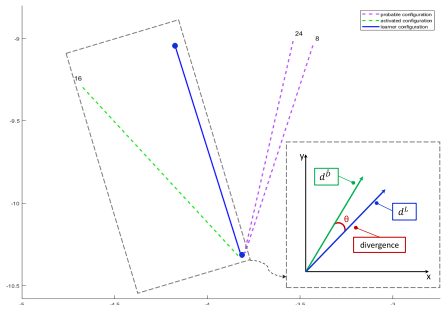


FIGURE 15. An example illustrates when the learner agent is in the exploration mode. The purple lines show the relative distance vector from the most probable configuration, while the green line is the relative distance from the activated configuration. The learner takes an action to explore the environment because the divergence between learner configuration and the activated one is more than ρ.

When L enters the exploration stage in a certain episode, it saves all the new explored configurations along with the performed actions. Then, L clusters those saved pairs (i.e., new configurations and actions) as discussed in Section III-B3. The new 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. Fig. 17 and Fig. 18 describe the clustering process of the new configurations in two scenarios related to lane-keeping and lane-changing. In each step, the newly learned clusters are appended incrementally to the model to modify and improve the action selection by exploiting new appended actions through the online learning phase and resolving the L's uncertainty about the surrounding environment. Fig. 19 and Fig. 21 illustrate the clusters of the reference FP model (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

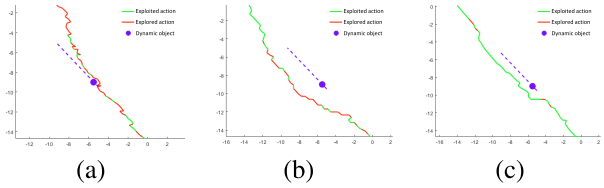


FIGURE 16. This figure shows three trajectories in different times of online learning. In (a), the learner experiences the new actions by exploration. By balancing the exploration and exploitation, the learner improves the action selection (b), and (c) shows the learner can decrease the explored action and make suitable decisions concerning the dynamic object.

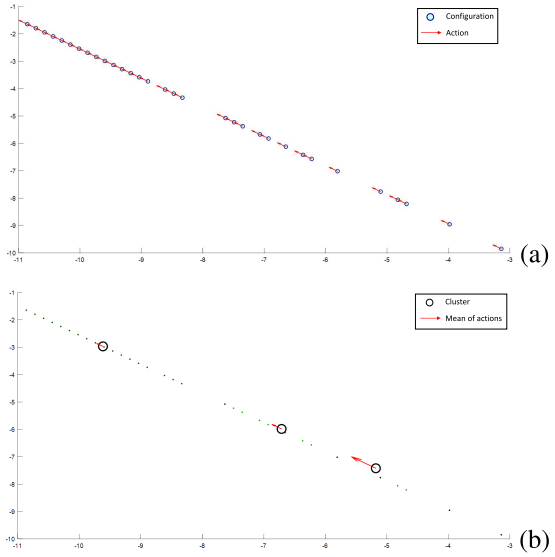


FIGURE 17. Clustering the explored configurations in lane-keeping scenario. a) It shows all-new exploration by the learner through one step, and b) it shows the clustered configurations and the corresponding mean action value to them (red arrows).

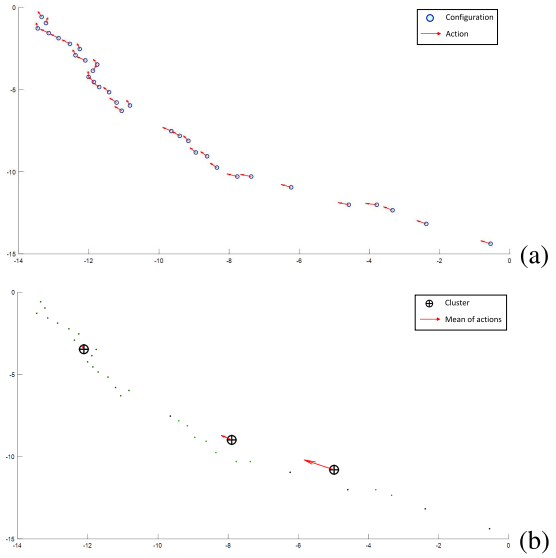


FIGURE 18. Clustering the explored configurations in lane-changing scenario. a) It shows all-new exploration by the learner through one step, and b) it shows the clustered configurations and the corresponding mean action value to them (red arrows).

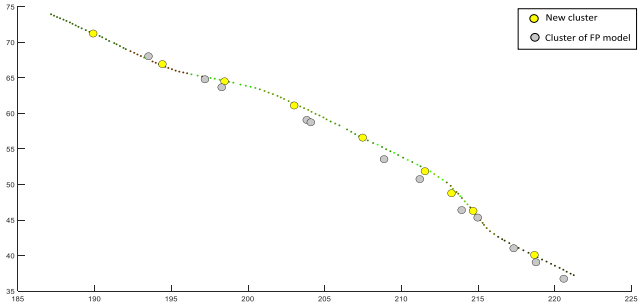


FIGURE 19. This figure shows the incremental learning of the model through the online phase. The gray circles show the clusters that belong to the FP model, and the yellow circles present the newly added clusters to the AFP model, which is learned through the exploration.

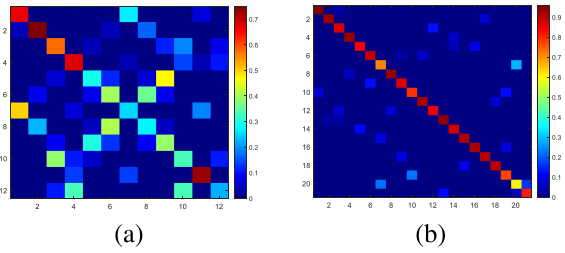


FIGURE 20. This figure is based on Fig. 19. a) Describes the TM related to the FP model in case of lane-changing from the right side that includes 12 cells, and b) shows TM with 21 cells after learning a new set of clusters (yellow circles in Fig. 19) that explains how the number of clusters increases in each online learning step.

the dynamic object $\hat{O}$. The corresponding TMs are updated by adding new rows and columns that represent the new learned configurations as shown in Fig. 20(b) and Fig. 22(b) and the TMs of the reference FP model are shown in Fig. 20(a) and Fig. 22(a). Comparing sub-figures (a) and (b) in each figure (Fig. 20 and Fig. 22) shows how the TMs of the FP model 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 the active inference 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. 10-(b). Two factors directly affect the success of the learner travel in each episode: the cumulative probability of going out of boundary and the collision probability. As we mentioned earlier, each episode includes ten steps (ten full paths). Obviously, with two factors decreasing at each step, the growth of the success steps led to an increase of the cumulative successful travel in each episode. By way of explanation, during the exploration, the model minimizes the FE measurement at the discrete level ($\ddot{F}$) at time k, which causes the resemblance between predictions and evidence at the continuous level. In total, by optimizing the global FE defined in (29) in the unseen situations, the learner can

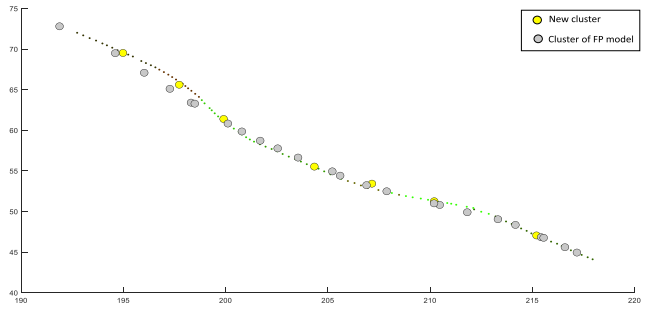


FIGURE 21. This figure shows another example of the incremental learning of the model through the online phase. The gray circles show the clusters that belong to the FP model, and the yellow circles present the newly added clusters to the AFP model, which is learned through the exploration.

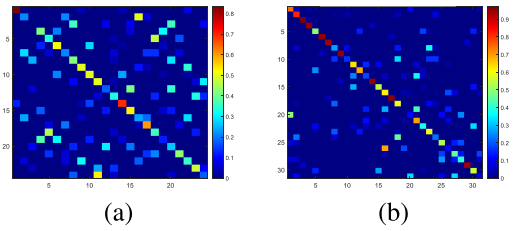


FIGURE 22. This figure is based on Fig. 21. a) Describes the TM related to the FP model in case of lane-changing from the left side that includes 24 cells, and b) shows TM with 31 cells after learning a new set of clusters (yellow circles in Fig. 21) that explains how the number of clusters increases in each online learning step.

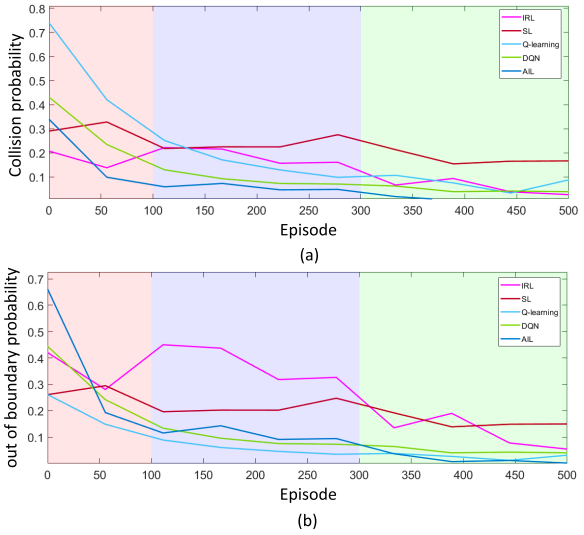


FIGURE 23. Analysis of the learning process. a) Collision probability curves, and b) going out of boundary curves.

manage to avoid a collision with another agent or going out of boundary.

Fig. 23-(a) shows the cumulative collision probabilities in each episode. We observe that the collision probability decreases as the number of episodes increases. Fig. 23-(b) presents the cumulative probabilities of going out of boundary that starts with 62% and during the learning dramatically

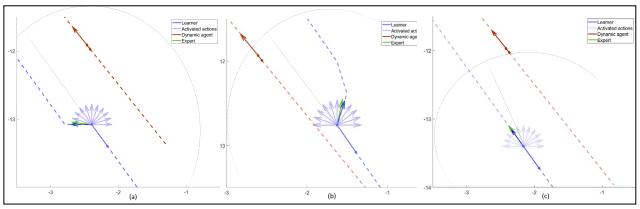


FIGURE 24. In three cases the learner agent's preference (blue arrow) is similar to the expert behavior (green arrow) in the same situation. Also the agent has learnt to keep safe distance (gray dashed line) with the another dynamic agent.

TABLE 4. This table shows the probability of actions selection by learner agent in Fig. 24 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, which has 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

declines to 0%. Fig. 23-(a)-(b) justifies the L behavior in Fig. 10-(b). The experimental results demonstrate that the proposed method enabled L to learn better driving skills than other RL methods. Integrating IL with IRL gives L a prior driving experience, which accelerates the learning rate and improves 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 discrete and continuous levels to minimise the distinction between the situation model and AFP model.

Furthermore, qualitative results show the ability to manage critical situations. Fig. 24 shows some representative cases of different scenarios. The 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 are depicted in Table. 4. In each case of decision (lane-keeping, change left, and change right), the most likely motion to the expert is selected, which has the highest probability than other candidates. Table. 4 shows the probability percentage of the activated actions in all three cases.

3) IMITATION LOSS

Our goal is to find the best set of actions that minimize the imitation loss in terms of FE. Fig. 25 shows that the normalized global FE ($\mathcal{G}$) drops down capably in less than 50 training episodes, and after 200 episodes, its value continues to decrease below 0.1. Moreover, Fig. 27 shows the $\mathcal{G}$ performance considering different L's preference, i.e., to keep following the other dynamic agent $\hat{O}$, 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 L agent. Fig. 28 illustrates the imitation

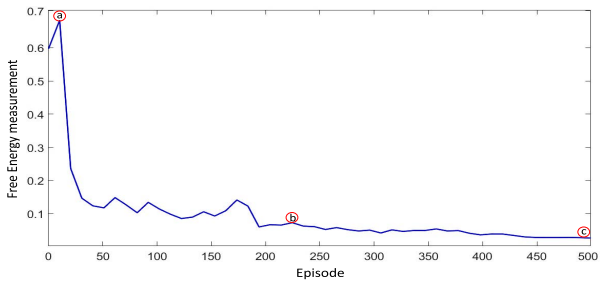


FIGURE 25. Global Free Energy measurement $\mathcal{G}$. The red circles show the FE measurement through three slots of learning: a) is at the beginning of training when the learner tries to experience the new action, b) the FE is declined cause improving the action selection, and at c) learner could decrease the distinction with the expert configurations. Fig. 26 show the three trajectories based on the mentioned measurements.

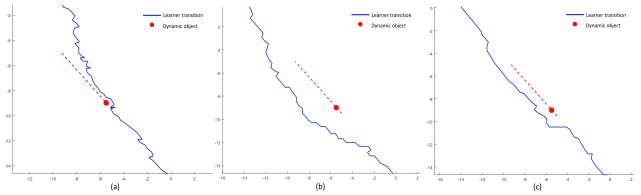


FIGURE 26. This figure shows three trajectories based on the selected FE measurements in Fig. 25. In (a), the learner can not balance the exploration and exploitation yet. By decreasing imitation loss and improving the explored actions, the learner can finish the travel by taking less actions (b), and (c) shows a successful travel with the suitable actions concerning the dynamic object's situation.

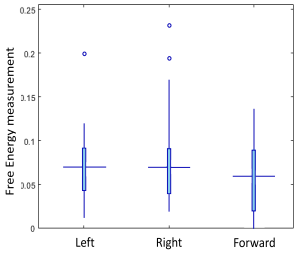


FIGURE 27. Free Energy measurement in three cases: lane-changing from left, lane-changing from right, and lane-keeping.

loss during the online active learning phase. We prove that our method can minimize the motion distinction ($\ddot{F}$), which is under control of action selection at each time instant. Further, improving the action selection process leads to minimizing the divergence ($\dot{F}$) 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. 28 shows that the proposed AIL is capable of achieving higher imitation rates compared with other learning methods.

Fig. 29 and Fig. 30 present 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. 29 and Fig. 30 provide comparison with other methods. It is shown that the proposed method (AIL) performs best among all methods (during training and testing),

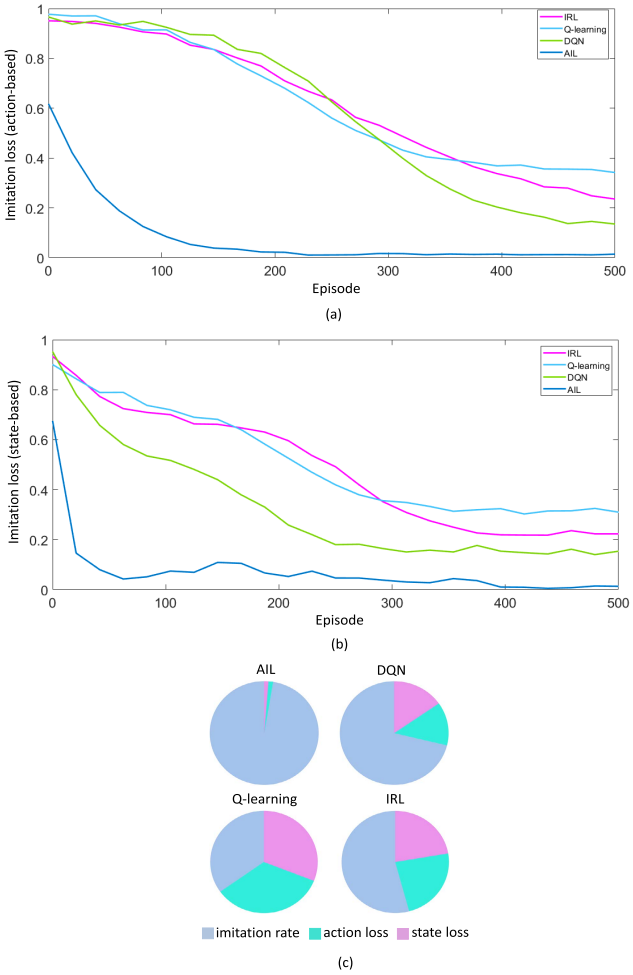


FIGURE 28. Analysis of the imitation process after 500 training episodes (5k path). a) Motion distinction. This figure shows the motion difference between the learner and the expert agent through the online learning active learning phase at time k. b) Divergence measurement. This figure shows the divergence between the learner and expert agent state after taking an action at time $k+1$, and c) Imitation loss.

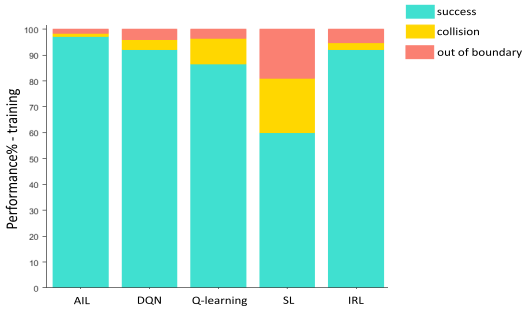


FIGURE 29. The training results after 500 episodes. AIL has less training loss (collision and going out of boundary) than other methods that it causes more training success percentage.

which is attributed to the effectiveness of the decision making while dealing with dynamic changes in the environment that improve the success rate by preventing going out of boundary and avoiding collisions.

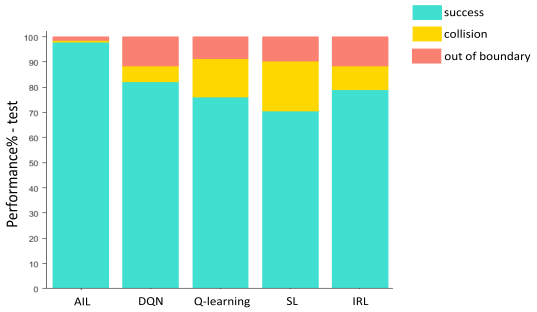


FIGURE 30. The testing results through the 500 paths. The testing path has different start positions than the training, and the dynamic object moves with different velocities during the training phase. It shows that the trained agent by AIL can achieve a high success percentage in the new environment.

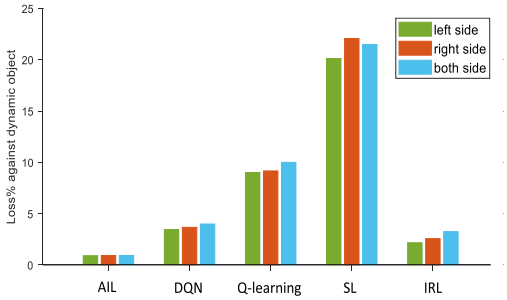


FIGURE 31. The result of overtaking loss from a dynamic object during changing-line from different side of the dynamic object.

TABLE 5. Results after 500 training episodes. * In SL method there is 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

Besides, the testing stage considers three scenarios: i) L tries to overtake $\hat{O}$ from the left side, ii) L tries to overtake $\hat{O}$ from the right side, and iii) L faces a mixed situation where it must overtake $\hat{O}$ from both the left and right sides. During testing, results showed that by $5k$ training episodes, the agent can change-lane to overtake the other dynamic agent in the environment effectively while other methods still have high collision probabilities as shown in Fig. 31. Correspondingly, Table. 4 summarises the performance metrics and present the comparison with other methods.

V. CONCLUSION

In this work, we proposed a novel framework to integrate Active Inference with Imitation Learning (i.e., AIL) for autonomous driving. The proposed AIL framework is based on learning a situation model encoded in a coupled Dynamic Bayesian Network (DBN) explaining the dynamic interactions between two moving agents (i.e., an expert agent and a dynamic object). The situation model is used to initialize a first-person (FP) 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-FP model consisting of the FP 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 behaviour 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 optimise 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 the 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 FE policies and conduct 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 (RL) methods such as Q-learning, Double Q-learning (DQN) and Inverse RL (IRL) in terms of the number of selected actions, successful travel rate, collision probability, out of boundary probability, and imitation loss. Future work will focus on integrating the Generalized Filtering on the Active-FP model to better utilize the updated transition matrices and improve predictive abilities at multiple levels that endow the learner agent with the capability to explain abnormal situations and how it can be avoided in the future.

REFERENCES

[1] W. Jiang, J. Lian, M. Shen, and L. Zhang, "A multi-period analysis of taxi drivers' behaviors based on GPS trajectories," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.

[2] M. Baydoun, M. Ravanbakhsh, D. Campo, P. Marin, D. Martin, L. Marcenaro, A. Cavallaro, and C. S. Regazzoni, "A multi-perspective approach to anomaly detection for self -Aware embodied agents," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 6598–6602.

[3] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 613–626, Feb. 2018.

[4] D. Fassbender, B. C. Heinrich, T. Luettel, and H.-J. Wuensche, "An optimization approach to trajectory generation for autonomous vehicle following," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 3675–3680.

[5] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, and I. Matthews, "Win at home and draw away: Automatic formation analysis highlighting the differences in home and away team behaviors," in *Proc. 8th Annu. MIT Sloan Sports Anal. Conf.*, 2014, pp. 1–7.

[6] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends Cognit. Sci.*, vol. 3, no. 6, pp. 233–242, Jun. 1999.

[7] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Survey: Robot programming by demonstration," Springrer, Tech. Rep., 2008.

[8] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Phil. Trans. Roy. Soc. London B, Biol. Sci.*, vol. 358, pp. 537–547, Apr. 2003.

[9] S. Schaal, "Learning from demonstration," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 1040–1046.

[10] R. Bellman, "A Markovian decision process," *Indiana Univ. Math. J.*, vol. 6, no. 4, pp. 679–684, Apr. 1957.

[11] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.

[12] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 204–211.

[13] S. Ross and J. A. Bagnell, "Reinforcement and imitation learning via interactive no-regret learning," 2014, *arXiv:1406.5979*.

[14] K.-W. Chang, A. Krishnamurthy, A. Agarwal, H. Daume, and J. Langford, "Learning to search better than your teacher," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2058–2066.

[15] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 6292–6299.

[16] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," 2017, *arXiv:1709.10087*.

[17] W. Sun, J. Andrew Bagnell, and B. Boots, "Truncated horizon policy search: Combining reinforcement learning imitation learning," 2018, *arXiv:1805.11240*.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and S. Petersen, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[20] A. Edwards, C. Isbell, and A. Takanishi, "Perceptual reward functions," 2016, *arXiv:1608.03824*.

[21] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *Proc. ICML*, vol. 1, 2000, p. 2.

[22] T. Munzer, B. Piot, M. Geist, O. Pietquin, and M. Lopes, "Inverse reinforcement learning in relational domains," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 1–8.

[23] D. Verma and R. P. Rao, "Goal-based imitation as probabilistic inference over graphical models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1393–1400.

[24] K. J. Friston, J. Daunizeau, and S. J. Kiebel, "Reinforcement learning or active inference," *PLoS ONE*, vol. 4, no. 7, 2009, Art. no. e6421.

[25] K. Friston, J. Kilner, and L. Harrison, "A free energy principle for the brain," *J. Physiol.*, vol. 100, nos. 1–3, pp. 70–87, 2006.

[26] K. Friston, "The free-energy principle: A rough guide to the brain?" *Trends Cognit. Sci.*, vol. 13, no. 7, pp. 293–301, Jul. 2009.

[27] H. Brown, K. Friston, and S. Bestmann, "Active inference, attention, and motor preparation," *Frontiers Psychol.*, vol. 2, p. 218, Sep. 2011.

[28] T. Gangwani and J. Peng, "State-only imitation with transition dynamics mismatch," 2020, *arXiv:2002.11879*.

[29] D. Brown, R. Coleman, R. Srinivasan, and S. Niekum, "Safe imitation learning via fast Bayesian reward inference from preferences," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1165–1177.

[30] M. Liu, T. He, M. Xu, and W. Zhang, "Energy-based imitation learning," 2020, *arXiv:2004.09395*.

[31] S. K. S. Ghasemipour, R. Zemel, and S. Gu, "A divergence minimization perspective on imitation learning methods," in *Proc. Conf. Robot Learn.*, 2020, pp. 1259–1277.

[32] R. Bhattacharyya, B. Wulfe, D. Phillips, A. Kuefler, J. Morton, R. Senanayake, and M. Kochenderfer, "Modeling human driving behavior through generative adversarial imitation learning," 2020, *arXiv:2006.06412*.

[33] D. Vogt, H. B. Amor, E. Berger, and B. Jung, "Learning two-person interaction models for responsive synthetic humanoids," *J. Virtual Reality Broadcastings*, vol. 11, no. 1, pp. 1–11, 2014.

[34] A. Droniou, S. Ivaldi, and O. Sigaud, "Learning a repertoire of actions with deep neural networks," in *Proc. 4th Int. Conf. Develop. Learn. Epigenetic Robot.*, Oct. 2014, pp. 229–234.

[35] M. Liu, W. Buntine, and G. Haffari, "Learning how to actively learn: A deep imitation learning approach," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2018, pp. 1874–1883.

[36] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auto. Syst.*, vol. 57, no. 5, pp. 469–483, 2009.

[37] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 661–668.

[38] M. Toussaint, "Robot trajectory optimization using approximate inference," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 1049–1056.

[39] H. J. Kappen, V. Gómez, and M. Opper, "Optimal control as a graphical model inference problem," *Mach. Learn.*, vol. 87, no. 2, pp. 159–182, May 2012.

[40] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. AAAI*, Chicago, IL, USA, vol. 8, 2008, pp. 1433–1438.

[41] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[42] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," 2018, *arXiv:1805.00909*.

[43] Q. Zhang, J. Lin, Q. Sha, B. He, and G. Li, "Deep interactive reinforcement learning for path following of autonomous underwater vehicle," *IEEE Access*, vol. 8, pp. 24258–24268, 2020.

[44] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10357–10377, 2021.

[45] A. Boularias, J. Kober, and J. Peters, "Relative entropy inverse reinforcement learning," in *Proc. 40th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 182–189.

[46] Y. Du and I. Mordatch, "Implicit generation and modeling with energy based models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32. Red Hook, NY, USA: Curran Associates, 2019, pp. 3608–3618.

[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[48] K. Friston, T. FitzGerald, F. Rigoli, P. Schwartenbeck, and G. Pezzulo, "Active inference and learning," *Neurosci. Biobehav. Rev.*, vol. 68, pp. 862–879, Sep. 2016.

[49] L. E. Sucar, "Probabilistic graphical models," in *Advances in Computer Vision and Pattern Recognition*, vol. 10. London, U.K.: Springer, 2015, pp. 1–978.

[50] Z. Ghahramani, "Learning dynamic Bayesian networks," in *International School on Neural Networks, Initiated by IIASS and EMFCSC*. Cham, Switzerland: Springer, 1997, pp. 168–197.

[51] D. J. Hill, B. S. Minsker, and E. Amir, "Real-time Bayesian anomaly detection for environmental sensor data," in *Proc. Congr.-Int. Assoc. Hydraulic Res.*, vol. 32, 2007, p. 503.

[52] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. New York, NY, USA: Academic, 1982.

[53] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic Bayesian networks," 2013, *arXiv:1301.3853*.

[54] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni, "Learning switching models for abnormality detection for autonomous driving," in *Proc. 21st Int. Conf. Inf. Fusion (FUSION)*, Jul. 2018, pp. 2606–2613.

[55] Y. Zheng, S. Jia, Z. Yu, T. Huang, J. K. Liu, and Y. Tian, "Probabilistic inference of binary Markov random fields in spiking neural networks through mean-field approximation," *Neural Netw.*, vol. 126, pp. 42–51, Jun. 2020.

[56] M. Baydoun, D. Campo, D. Kanapram, L. Marcenaro, and C. S. Regazzoni, "Prediction of multi-target dynamics using discrete descriptors: An interactive approach," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3342–3346.

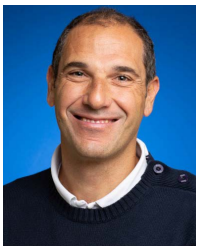
[57] K. Friston, B. Sengupta, and G. Auletta, "Cognitive dynamics: From attractors to active inference," *Proc. IEEE*, vol. 102, no. 4, pp. 427–445, Apr. 2014.

[58] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*. Chapel Hill, NC, USA, 1995.
[59] H. Iqbal, D. Campo, M. Baydoun, L. Marcenaro, D. M. Gomez, and C. Regazzoni, "Clustering optimization for abnormality detection in semi-autonomous systems," in *Proc. 1st Int. Workshop Multimodal Understand. Learn. Embodied Appl. (MULEA)*, 2019, pp. 33–41.
[60] K. Friston, T. FitzGerald, F. Rigoli, P. Schwartenbeck, and G. Pezzulo, "Active inference: A process theory," *Neural Comput.*, vol. 29, no. 1, pp. 1–49, 2017.
[61] K. Friston, F. Rigoli, D. Ognibene, C. Mathys, T. Fitzgerald, and G. Pezzulo, "Active inference and epistemic value," *Cognit. Neurosci.*, vol. 6, no. 4, pp. 187–214, 2015.
[62] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.
[63] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The Mahalanobis distance," *Chemometrics Intell. Lab. Syst.*, vol. 50, no. 1, pp. 1–18, Jan. 2000.
[64] P. Marín-Plaza, J. Beltrán, A. Hussein, B. Musleh, D. Martín, A. de la Escalera, and J. M. Armingol, "Stereo vision-based local occupancy grid map for autonomous navigation in ROS," in *Proc. 11th Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2016, pp. 1–6.

SHEIDA NOZARI (Member, IEEE) the master's degree in computer engineering-artificial intelligence from Azad University, Iran, in 2017, and the master's degree in human–computer interaction from the University of Paris-Saclay, France, in 2019. She is currently pursuing the Ph.D. degree with the Joint Doctorate in Self-Aware Autonomous System Program between the University of Genoa, Italy, and the Carlos III University of Madrid, Spain. Her research interests include imitation learning, self-awareness, and autonomous multi-agent and deep learning techniques for cognitive and interactive environments.

ALI KRAYANI (Member, IEEE) received the B.S. degree in telecommunication engineering from the Politecnico di Torino, Italy, in 2014, the M.S. degree in telecommunication engineering from the University of Florence, Italy, in 2017, and the joint Ph.D. degree from the Joint Doctorate in Interactive and Cognitive Environments Program between the University of Genoa, Italy, and the Queen Mary University of London, U.K., in April 2022. He worked as a Software Engineer for several companies and he is currently a Postdoctoral Research Fellow with the Electrical, Electronics and Telecommunication Engineering and Naval Architecture Department (DITEN), University of Genoa. His current research interests include cognitive and AI-enabled radios, wireless communications, UAV communications, self-awareness, dynamic Bayesian networks, active inference, and artificial intelligence.

PABLO MARIN-PLAZA received the degree in industrial electronics and automation engineering from the Universidad Carlos III de Madrid, in 2011, and the master's degree in robotics and automation, in 2013. In 2012, he joined the Department of Systems and Automation Engineering, Universidad Carlos III de Madrid, becoming a member of the Intelligent Systems Laboratory. He started working as an Assistant Lecturer, in 2013. He is currently working on several projects attached to the University Carlos III related to autonomous vehicles and computer vision. His current research interests include computer vision and autonomous ground vehicles. In 2018, he obtained the title of Doctor Engineering in Electric, Electronic, and Automation.



LUCIO MARCENARO (Senior Member, IEEE) is currently an Associate Professor of telecommunications at the University of Genoa, Italy. He has over 20 years of experience in image and video sequence analysis. He has authored about 160 scientific articles on signal and video processing for computer vision. He is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, a Technical Program Co-Chair of the 13th International Conference on Distributed Smart Cameras (ICDSC) and for the first IEEE International Conference on Autonomous Systems (IEEE ICAS 2021), a Co-Organizer of the 2019 Summer School on Signal Processing (S3P), and the General Chair of the Symposium on Signal Processing for Understanding Crowd Dynamics. He is active within the IEEE Signal Processing Italy Chapter and the Director of Student Services Committee (2018–2021).

DAVID MARTÍN GÓMEZ (Member, IEEE) graduated in industrial physics (automation). He received the Ph.D. degree in computer science, in 2008. He is currently an Associate Professor and a Senior Researcher at the Carlos III University of Madrid and a member of the Intelligent Systems Laboratory. His main research interests include computer vision, sensor fusion, intelligent transportation systems, advanced driver assistance systems, autonomous ground vehicles, unmanned aerial vehicles, vehicle positioning and navigation, and reasoning and decision-making under uncertainty in autonomous vehicles. He is a member of the Spanish Computer Vision Group, a member of the IEEE Intelligent Transportation Systems Society, a member of the IEEE Signal Processing Society, and a member of the IEEE Autonomous Systems Initiative (SPS-ASI).

CARLO REGAZZONI (Senior Member, IEEE) is currently a Full Professor of cognitive telecommunications systems at the DITEN, University of Genoa, Italy. He has been responsible of several national and EU funded research projects. He is also the Co-ordinator of international Ph.D. courses on interactive and cognitive environments involving several European universities. He is the author/coauthor of more than 100 articles on international scientific journals and of more than 300 papers at peer-reviewed international conferences. He served as the general chair in several conferences and an associate/guest editor in several international technical journals. He has served in many roles in governance bodies of IEEE SPS. From 2015 to 2017, he was serving as the Vice President Conferences IEEE Signal Processing Society.